



● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

Hanttu, Piia

2011 Leppävaara

Laurea-ammattikorkeakoulu
Laurea Leppävaara

Tietojärjestelmän tuki- ja ylläpitokäytäntöjen parantaminen

Hanttu, Piia
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Toukokuu 2011

Hanttu, Piia

Tietojärjestelmän tuki- ja ylläpitokäytäntöjen parantaminen

Vuosi	2011	Sivumäärä	55
-------	------	-----------	----

Tämä opinnäytetyö käsittelee ylläpitokäytäntöjen parantamista käytössä olevalle toiminnanohjausjärjestelmälle. Ylläpitokäytännöt ovat jaettu kahteen osaan; järjestelmän tukeen ja järjestelmän pienkehitykseen. Järjestelmän tukikäytännöt kattavat yrityksen palvelupisteen osalta toiminnanohjausjärjestelmän tukipyyntöjen tehokkaan käsittelyn sekä oikeanlaisen tukipyyntöjen reitityksen järjestelmän tukihenkilöiden osalta, josta lopputuotoksena tehdään tukipyynnön käsittelyprosessikuvio. Tukikäytäntöjen kehittämiseen sisältyi lisäksi käyttäjille suunnatun tukikuvion tekeminen, jossa liiketoimintaresurssit on otettu tukipyyntöprosessiin mukaan. Järjestelmän pienkehitystä varten on vertailtu eri menetelmiä pienkehitystehtävien järjestelmälliseen ja hallittuun toteuttamiseen sekä mietitty näistä menetelmistä hyviä ominaisuuksia kyseisen järjestelmän pienkehitykseen.

Ensisijaisena tavoitteena opinnäytetyössä oli parantaa tuotannonohjausjärjestelmän käyttäjätyytyväisyyttä sekä luottamusta tukiprosessiin ja tehostaa tukipyyntöjen kulkua tukiprosessissa ottaen siihen mukaan kaikki mahdolliset resurssit. Toissijaisena tavoitteena oli etsiä sopiva menetelmä pienkehitystehtäviin. Tukiprosessin kehittämiseen sekä pienkehitysmenetelmän löytämiseksi sovellettiin ennalta valittuja teorioita, jotka olivat ITIL sekä Agile-menetelmiin kuuluvat Scrum ja Kanban.

Opinnäytetyö toteutettiin toiminnallisen tutkimuksen näkökulmasta. Työssä käytettiin lähtötilanteen selvittämiseksi havainnointia, vapaamuotoista haastattelua sekä käyttäjätyytyväisyyskysely. Työn tuotoksina on uudistettu tukiprosessi loppukäyttäjän sekä palvelupisteen näkökulmasta, kyselylomake järjestelmän tukipyyntöihin, muutospyyntölomake pienkehitystehtäviin sekä Agile-menetelmien vertailun tulokset. Työn vaikutusta järjestelmän käyttäjätyytyväisyyteen mitataan vertailemalla käyttäjätyytyväisyyskyselyn tuloksia keskenään.

Jatkokehittämisalueina ovat pienkehitystehtävien hallinnointiprosessin suunnittelu, kysely- ja muutospyyntölomakkeen kehittäminen sekä tyytyväisyyskyselyn toteutussuunnitelman laatiminen ja sen hyödyntämissuunnitelma.

Hanttu, Piia

Improving a support and maintenance practises for an information system

Year	2011	Pages	55
------	------	-------	----

This thesis describes a maintenance process for an existing enterprise resource planning system. The maintenance process will consist of two parts, support and small development. The support process covers all the service requests and incidents regarding this enterprise's resource planning system, which will be handled by the Service Desk. The objective of a support process is to handle incidents and service requests in an efficient and correct manner. The support process also includes developing a support process from the user point of view where right resources are included, as well as a process for the Service Desk how to handle incident and service requests using the right resources. The small development objective was to compare different methods in order to make small development more controlled.

The primary purpose of this thesis was to improve the support process by optimizing the handling of incident and service requests, including right resources and improving customer satisfaction and confidence in the support process. The secondary purpose was to compare the Agile methods for the small development process. The theories used to examine the support process and small development process chosen beforehand are ITIL and the Agile methods Scrum and Kanban.

The research perspective of this thesis was action research. Observation, inquiries and unofficial interviews were used to analyse the baseline of the research. The results of the thesis are an improved support process from the user and service desk point of view, a questionnaire form for systems incident and service requests, a change request form for small development ideas, and the results of the Agile methods comparisons. The indicator of the effectiveness of the assignment is the comparison of the customer satisfaction survey.

Themes for further research are planning a process for small development team, developing questionnaire form, change request form and as well as an action plan and an exploitation plan for customer satisfaction surveys.

Key words action research, ITIL, maintenance, service desk, support process, system support

Sisällys

1	Johdanto	6
1.1	Kohdeyritys	6
1.2	Rajaus	7
1.3	Lähtötilanne kehitystyölle - syksy 2010	7
1.4	Toiminnallinen opinnäytetyö	9
1.5	Tutkimuksen toteutus	10
1.6	Tutkimusaineiston keruu	10
1.7	Kyselylomakkeen laadinta	10
1.8	Tutkimustulosten luotettavuus	11
2	Kehitystyön tavoite	13
2.1	Palvelupisteen toiminnan parantaminen	13
2.2	Pienkehityskäytäntöjen parantaminen	14
3	Keskeiset käsitteet	15
3.1	Information Technology Infrastructure Library	15
3.2	Palvelupiste	16
3.3	Palvelutasosopimus (SLA)	17
3.4	Prosessi	17
3.5	Tietojärjestelmä	17
3.6	Toimintatutkimus	18
4	Ylläpito	18
4.1	Ylläpidon tarve	18
4.2	Ylläpidon osa-alueet	19
4.3	Ylläpitoon liittyvät ongelmat	20
4.4	Muutosten hallinta	23
5	Kanban	23
5.1	Kanbanin periaatteet	24
5.2	Kanbanin määrittelemä työskentelymalli	25
6	Scrum	25
6.1	Scrumin määrittelemät roolit	27
6.2	Scrumin mukainen työskentely	27
7	Palvelun toimittaminen	28
7.1	Tapahtumanhallinta	28
7.2	Palvelupyyntöprosessi	31
8	Kehitystoimenpiteet	33
8.1	Järjestelmän tuki	33
8.1.1	Tukipyyntöprosessi	33
8.1.2	Tukipyyntöjen käsittelyprosessi	34
8.1.3	Kyselylomake	35

8.1.4	Yleiset ohjeistukset	35
8.2	Pienkehitys	36
8.2.1	Kanbanin ominaisuuksien arviointi	36
8.2.2	Scrumin ominaisuuksien arviointi.....	36
8.2.3	Yhteenvedo menetelmien soveltuvuudesta pienkehitykseen	37
8.2.4	Muutospyyntölomake	38
9	Kehitystyön tulosten arviointi	39
9.1	Käyttäjätyytyväisyyskyselyn tulosten vertailu	40
9.2	Työn onnistumisen arviointi	45
10	Pohdinta	46
10.1	Johtopäätökset	46
10.2	Oman ammatillisen kehityksen arviointi	47
10.3	Jatkokehittämisaalueet.....	47
	Kuviot	51
	Liite 1 A: Käyttäjätyytyväisyyskysely.....	52
	Liite 1 B: Käyttäjätyytyväisyyskysely, avoimet kysymykset	53
	Liite 2: Kysymyslomake tukipyyntöjen käsittelyyn	54
	Liite 3: Muutospyyntölomake	55

1 Johdanto

Tämän opinnäytetyön tarkoituksena on suunnitella käytössä olevalle toiminnanohjausjärjestelmälle ylläpidon tukiprosessi, joka kattaa 1) järjestelmän tukipyyntöjen käsittelyn sekä 2) ennalta valittujen menetelmien soveltuvuuden tutkimisen pienkehitysprosessiin, järjestelmän virheiden korjaamisen ja päivittämisen hallinnoimisen tueksi. Tämä opinnäytetyö on toteutettu toiminnallisen tutkimustavan näkökulmasta, joten työ on rakennettu tutkimussuunnitelman rakennetta noudattaen.

Luvussa 2 käydään lyhyesti läpi tutkimuksen toteutus. Luvussa 3 kuvataan ensimmäiseksi työn keskeiset tavoitteet. Tavoitteet määriteltiin yhdessä työn tilaajan kanssa. Ensisijaisena tavoitteena työssä on suunnitella tukiprosessin toiminta loppukäyttäjän näkökulmasta sekä tukipyyntöjen käsittelyprosessi palvelupisteelle, joka yrityksessä tunnetaan paremmin englanninkielisenä terminä service desk. Toissijaisena tavoitteena työlle on asetettu eri menetelmien tutkiminen järjestelmän pienkehitystehtäviin. Menetelmien vertailulla pyritään selvittämään niiden ominaisuuksien hyödyntäminen pienkehitystehtäviin sekä mahdollisesti näiden kahden eri Agile-menetelmän ominaisuuksien yhdistäminen.

Luvussa 4 on käsitelty keskeisimmät käsitteet, jotka liittyvät olennaisesti tähän työhön ja luvussa 5-8 on koostettuna keskeisimmät teoriat ja menetelmät, joita hyödyntäen ylläpidon tukiprosessia voidaan kehittää. Luvussa 9 on kuvattu teorian soveltaminen käytäntöön eli tämän työn tuotokset. Luvussa 10 on arvio toimenpiteiden onnistumisesta niin työn tekijän kuin työn tilaajan näkökulmasta ja lopuksi luvussa 11 on pohdittu työn tuloksellisuutta, joka sisältää johtopäätökset tästä kehitystyöstä, jossa arvioidaan työn tekijän ammatillista kasvua sekä pohditaan mahdollisia jatkotoimenpiteitä.

1.1 Kohdeyritys

Yritys on perustettu vuonna 1800-luvulla, ja se kuuluu maailman kymmenen suurimman rakennusyrityksen joukkoon ja on yksi maailman 500 isoimmasta yrityksestä. Yrityksen toimialueisiin kuuluu Eurooppa, Yhdysvallat ja Latinalainen Amerikka. Yrityksen toimialoihin kuuluu maakohtaiset rakentamispalvelut sekä kansainvälinen asuntojen, toimitilojen ja infrastruktuurin projektikehitys. Suomessa yrityksellä työskentelee noin 2200 henkilöä, joista Suomen tietohallinnossa työskentelee noin 15 henkilöä, jotka vastaavat pääosin IT-toimintojen toimivuudesta ja toimittamisesta Suomen liiketoiminnan tarpeisiin.

1.2 Rajaus

Tämä opinnäytetyö keskittyy ITIL v3:n palvelun toimittaminen - osa-alueeseen. Muita ITIL v3:n osa-alueita ei käsitellä tämän työn puitteissa. Palvelun toimittaminen liittyy hyvin läheisesti yrityksen päivittäiseen toimintaan järjestelmien tukemisen kautta, joka on tämän työn tärkein kehittämiskohde. ITIL v3:sta sekä sen palvelun toimittaminen osa-alueesta on kerrottu tarkemmin luvussa 4 ja 8, jossa selitetään tästä osa-alueesta valitut prosessit, joita olivat tapahtumanhallinta sekä palvelupyyntöprosessi. Aiheen ulkopuolelle rajataan järjestelmän osien muuttaminen tai uusien ominaisuuksien lisääminen, joiden tekeminen luokitellaan yrityksen standardien mukaan projektiksi eli joiden kesto on yli seitsemän päivää tai kustannukset ylittävät kokonaisuudessa 50 000 euroa. Tässä opinnäytetyössä ei myöskään oteta kantaa järjestelmän ylläpidon budjettiin, dokumentointiin, ohjelmointikieleen, tietoturvaan eikä muutosten viestintäprosessiin. Aiheen ulkopuolelle rajataan myös ylläpidossa käytettävät työkalut ja ohjelmistot sekä muutosten suunnitteluprosessi.

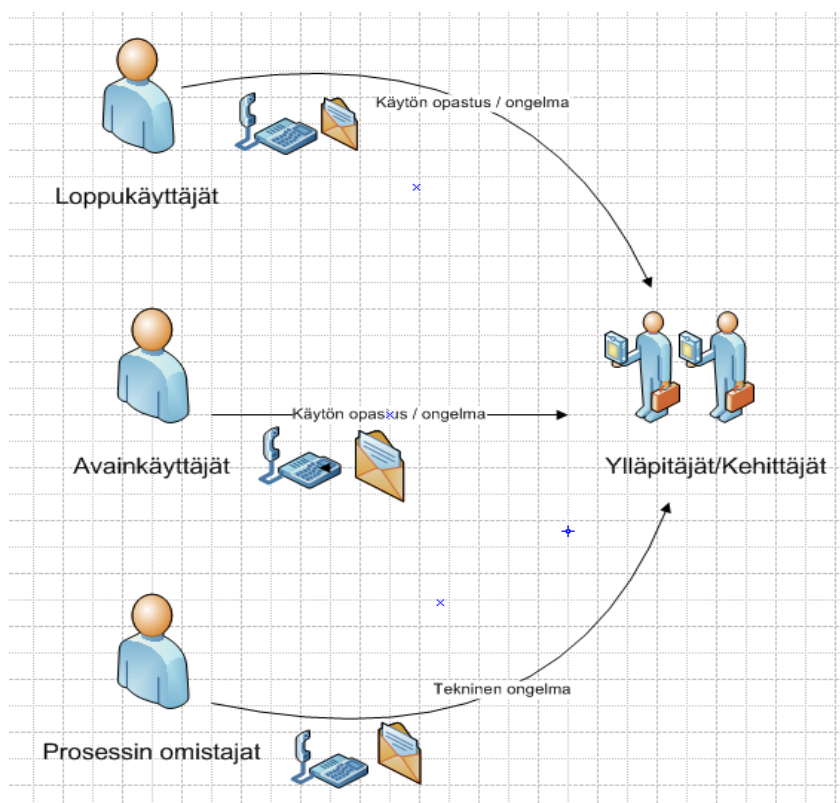
1.3 Lähtötilanne kehitystyölle - syksy 2010

Tietojen luottamuksellisuudesta johtuen järjestelmän täsmällistä käyttötarkoitusta tai käyttäjämäärää/ryhmiä ei ole kuvattu tässä opinnäytetyön versiossa.

Käyttöönottovaiheessa järjestelmälle oli kehitetty tukiprosessi, jossa liiketoiminta on vastuussa järjestelmän käytön tukemisesta ja kehitysasioiden eteenpäin viemisestä. Ongelmana tukiprosessin toteutumisessa oli kuitenkin se, että kaikki tukiprosessiin mukaan otetut henkilöt eivät osanneet käyttää järjestelmää tarpeeksi hyvin, jotta he olisivat voineet antaa tukea sen käytössä. Myös ongelmana oli, että osa tukihenkilöistä ei välttämättä edes tiennyt olevansa tukihenkilö kyseiseen järjestelmään taikka olivat unohtaneet kyseisen roolin. Yhtenä suurimpana ongelmana järjestelmän tukiprosessissa kuitenkin mielestäni oli se, että loppukäyttäjiä ei informoitu tarpeeksi tästä tukiprosessista, jolloin loppukäyttäjät eivät tienneet oikeaa yhteyshenkilöä eri ongelmatilanteissa, vaan kysyivät neuvoa käytön ongelmissa sekä virheen korjausapua aina samalta henkilöltä.

Järjestelmän käytön yleistyessä käyttäjien tukipyyntöjen määrä kasvoi ja koska tuki- ja ylläpitoprosesseja ei viety aivan loppuun asti, suurin osa asiakkailta tulevista yhteydenotoista ohjautui prosessissa järjestelmän ylläpitäjien ja kehittäjien selvitettäväksi, koska käyttäjät olivat tottuneet saamaan ratkaisut IT-osastolta kaikissa tietotekniikkaan liittyvissä ongelmatilanteissa. Käyttäjien lisäksi järjestelmän avainkäyttäjät sekä osa prosessin omistajista kuormitti IT-osastolla työskenteleviä resursseja esiin tulleilla ongelmilla sekä kehitysehdotuksilla. Tästä seurasi edellä mainittujen resurssien ylikuormitus, minkä seurauksena järjestelmän pieneki-

tystehtävien toteuttaminen sekä tukipyynnöjen ratkaisu kesti normaalia kauemmin. Ylläpidon lähtötilanteen havainnoimiseksi luotiin tukiprosessikuva (kuvio 1) pohjaksi kehitystyölle sekä luvussa 2 käyttäjätyytyväisyystulokset (kuvio 3) auttavat hahmottamaan lähtötason tukiprosessin parantamiselle. Näitä lähtötilannekuvioita käytetään mittareina työn loppuvaiheessa opinnäytetyön tuloksellisuuden arvioinnissa.



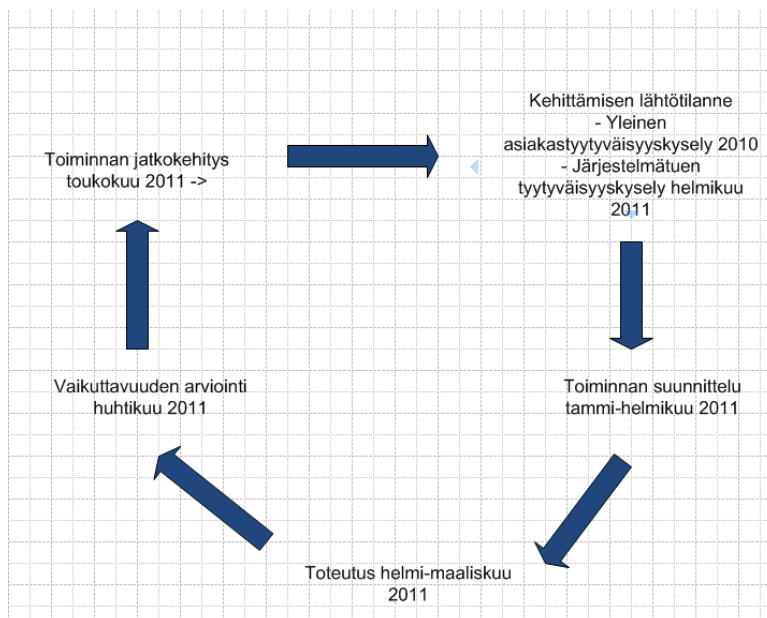
Kuvio 1. Lähtötilanne tukiprosessin kehittämiseksi syksyllä 2010

Työn lähtötilanne on kartoitettu havainnoimalla nykyistä tilannetta, haastattelemalla henkilöitä järjestelmän ylläpidossa ja myös muissa tehtävissä yrityksen IT-osastolla. Näiden lisäksi opinnäytetyön aikana liittyen järjestelmän tukipyynnöjen käsittelyyn, on toteutettu kysely (Liite 1). Kyselyn avulla pyritään kartoittamaan tämän kehitystyön vaikutusta tukipyynnöjen käsittelyn tehokkuuteen (Liite 1, A-osa). Lisäksi kyselyn avulla pyritään myös saamaan kyselyyn osallistujilta jatkokehitysehdotuksia järjestelmän toimintoihin sekä palvelupisteen toimintaan liittyen (Liite 1, B-osa). Työn aikana tehdyssä kyselyssä vertailtiin syksyn 2010 tilannetta nykyhetkeen, jolloin tukiprosessia on pyritty tehostamaan liiketoimintaresurssien avulla sekä oikeanlaisella tukipyynnöprosessilla. Nämä edellä mainitut toimenpiteet tehdään, jotta löydetään ylläpitoon mahdollisimman käytännöllinen ja tehokas menetelmä, jonka avulla pystytään toteuttamaan järjestelmän pienkehitystä hallitusti.

1.4 Toiminnallinen opinnäytetyö

Tässä opinnäytetyössä käytetään toimintatutkimuksen periaatetta työn eteenpäin viemisessä. Tämän työn tarkoituksena on muuttaa yrityksen ylläpitokäytäntöjä parempaan suuntaan tehostamalla tietojärjestelmän tukiprosessia sekä vertailemalla eri menetelmiä, joita on käytetty yleisesti järjestelmien ylläpidossa, jotta pienkehityksessä tulevien ominaisuuksien käyttöönotto olisi hallitumpaa ja selkeämpää. Toiminnallisen opinnäytetyön tavoitteena on käytännön toiminnan ohjeistaminen sekä järjeistämisen. (Airaksinen & Vilkkä 2003, 9-10.) Airaksinen & Vilkkä ovat myös maininneet kirjassaan toiminnalliselle opinnäytetyölle asetetut vaatimukset, joita ovat työelämälähtöisyys, käytännöllisyys, tutkimuksellisella asenteella toteutettu sekä alan tietojen ja taitojen hallintaa osoittava. Tämän työn toteuttaminen vaatii järjestelmän ylläpitoon liittyvien tehtävien tuntemista sekä ylläpidossa ja järjestelmätuessa yleisesti käytettyjen menetelmien, kuten ITIL v3:n sekä Agile-menetelmien tuntemusta sekä näiden ymmärrystä, jotta niitä voidaan soveltaa käytäntöön. Tämä työ vastaa myös Airaksisen & Vilkan vaatimuksiin työelämälähtöisyydestä ja käytännöllisyydestä, koska tämä työ tehdään puhtaasti yrityksen tarvetta mukaillen pyrkimyksenä tehostaa yrityksellä käytössä olevan järjestelmän ylläpitoprosessia.

Työn etenemisvaiheiden kuvaamiseksi suunniteltiin alla oleva prosessikuva (Kuvio 2), jonka tarkoituksena on antaa selkeämpi kuva tämän kehitystyön vaiheista sekä ilmaista lukijalle myös, että kehittäminen on jatkuva prosessi.



Kuvio 2. kehitystyön vaiheet

1.5 Tutkimuksen toteutus

Tutkimuksen lähtökohdan selvittämiseksi toteutettiin kysely, joka tunnetaan myös survey-tutkimuksen yleisimpänä menetelmänä. Kyselyssä aineisto kerätään vakioidusti ja siinä kysymykset ovat täysin samanlaiset kaikille kyselyyn vastaajille. Kyselyssä vastaajat muodostavat otoksen tai näytteen tietystä perusjoukosta. (Hirsjärvi ym. 2000, 180.)

Kysely lähetettiin osallistujille sähköpostilla helmikuussa 2011. Kyselyn tarkoituksena oli kartoittaa yrityksen palvelupisteen osuutta järjestelmän tukeen sekä samalla kartoittaa palvelupisteen osaamistasoa järjestelmän tukipyyntöihin liittyen. Kyselyn vastaamisaika oli kaksi viikkoa.

1.6 Tutkimusaineiston keruu

Tutkimuksen perusjoukko valittiin heidän tehtävärooliensa perusteella ja he olivat tässä tutkimuksessa järjestelmän tuessa toimivia henkilöitä. Perusjoukolla tarkoitetaan tutkimukseen valittua joukkoa henkilöitä, joista halutaan tehdä päätelmiä tutkimuksessa ja joista otantamenetelmän avulla valitaan otos, joka edustaa koko perusjoukkoa. Kokonaisotantaan valitaan koko perusjoukko, jota voidaan käyttää pienissä tutkimuksissa. (Vilkkä 2007, 51-52.)

Tämän opinnäytetyön tutkimusaineisto kerättiin kokonaisotantana yrityksen liiketoiminnan henkilöistä, jotka vastaavat kyseisen järjestelmän kehittämisestä sekä tukemisesta. Osa kyselyyn valituista henkilöistä siis vastaa kyseisen toiminnanohjausjärjestelmän kehittämisestä ja osa järjestelmän käytön opastamisesta. Kyselyn ulkopuolelle jätettiin tarkoituksella järjestelmän loppukäyttäjät, koska kyselyyn valitut henkilöt toimivat heidän tukenaan ja palvelupisteen tulisi olla näkyvä osa tulevaisuudessa vain järjestelmän tuessa toimiville henkilöille, ei loppukäyttäjille.

Kysely lähetettiin kaikille järjestelmän avainkäyttäjille sekä prosessien omistajille, eli yhteensä 56 henkilölle, joista kyselyyn vastasi 27 henkilöä, joten kyselyn vastausprosentti oli 48,2 ja katoprocentti 51,8. Kvantitatiiviset tulokset ovat tilastollisesti suuntaa antavia syksyn 2010 tilanteesta vähäisen osallistumisen vuoksi, mutta kysely oli ainoa keino saada selville syksyllä vallinneen tilanteen mielipiteet käyttäjiltä.

1.7 Kyselylomakkeen laadinta

Kysely jaettiin kahteen osaan, joista ensimmäinen osa (Liite 1 A) sisälsi lähtötilanteen selvittämiseen liittyvät kysymykset. Vastausvaihtoehdot olivat näissä kysymyksissä 1-4, jossa numero yksi (1) edusti huonoa tulosta ja numero neljä (4) hyvää tulosta. Tämä sama asteikko oli

kaikissa ensimmäisen osan (Liite 1 A) kysymyksissä, mutta kysymyksestä riippuen numeroa vastaava sanallinen vastaus oli erilainen mutta kuitenkin vastasi sen numeraalista arvoa. Kyselyn toinen osa (Liite 1 B) toteutettiin avoimilla kysymyksillä, joita ei käytetty lähtötilanteen selvittämiseen vaan ainoastaan jatkokehityksen avuksi.

Kyselyn ensimmäisessä osassa kartoitettiin viittä eri asiaa, jotka olivat helpdeskin osaamista, tukipyyntöjen vastausaika ja ratkaisuaika, ratkaisun saaminen helpdeskistä (palvelupisteestä) sekä tukipyynnöistä viestiminen. Kyselyssä ei ollut taustatietokysymyksiä, koska tutkimustuloksiin nähden kaikki vastaukset ovat tasavertaisia toisiinsa nähden. Kyselyn tarkoituksena oli kartoittaa järjestelmän tuessa toimivien henkilöiden mielipiteitä yrityksen palvelupisteen toimivuudesta ja osaamisesta järjestelmän tukipyyntöihin liittyen.

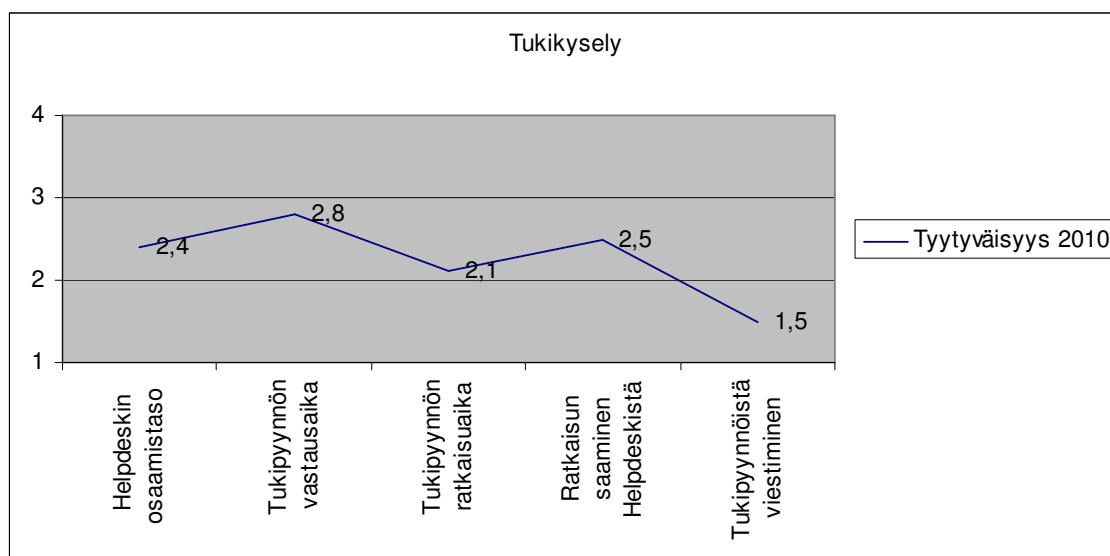
Kyselyn toisessa osassa (Liite 1 B) pyydettiin vastaajilta kehitysehdotuksia yrityksen tukipalveluun eli helpdeskiin liittyen sekä muuta palautetta liittyen järjestelmän kehitykseen. Näitä vastauksia tullaan hyödyntämään tämän opinnäytetyön jatkokehitystoimenpiteisiin.

1.8 Tutkimustulosten luotettavuus

Kyselylomakkeella pyritään vähentämään vastausvirheitä, joita syntyy helposti tutkijan tulkitessa haastatteluista saatuja vastaajien epätasällisiä tai epämääräisiä vastauksia. (Holopainen & Pulkkinen 2003, 39.) Kyselylomakkeen avulla pyritään siis saamaan mahdollisimman luotettava tutkimustulos. Kyselyyn otettiin mukaan strukturoituja kysymyksiä, jotta niiden analysointi olisi mahdollisimman luotettavaa eikä perustuisi sanallisten vastausten epätarkkaan tulkitsemiseen.

Useassa mielipidemittauksessa on tarkoituksena mitata muun muassa vastausten keskiarvoja. (Holopainen & Pulkkinen 2003, 13.) Tässäkin tutkimuksessa on tarve mitata vastausten keskiarvojen kautta niiden eroavaisuutta ja työn tuloksellisuutta. Vastausten analysointiin käytettiin Excel-ohjelmaa, jolla laskettiin vastauksien keskiarvo ja joiden perusteella rakennettiin taulukko, jossa viiva edustaa vastausten keskiarvoa, y-akseli ilmaisee vastausten arvosanat ja x-akseli näyttää kysymykset (Kuvio 3). Taulukosta nähdään helposti kysymysten vastauksen keskiarvo. Arvosanoista yksi (1) tarkoittaa huonoa tulosta ja neljä (4) hyvää tulosta. Taulukon avulla nähdään siis selkeästi tulosten vertailuvaiheessa, kuinka paljon käyttäjätyytyväisyyttä on onnistuttu kasvattamaan lähtötilanteeseen nähden, kun nykytilannetta taulukossa kuvaa erivärinen viiva.

Taulukossa (kuvio 3) kysymyksillä 1-4 oli kyselyssä vastausvaihtoehdot yhdestä neljään (1-4) ja tukipyynnöistä viestiminen -kysymyksellä oli arvoasteikko 1-2, joten sitä ei voi verrata muihin taulukossa oleviin kysymyksiin, mutta se otettiin kuitenkin taulukkoon mukaan jotta myös sen keskiarvoa pystytään vertailemaan helposti.



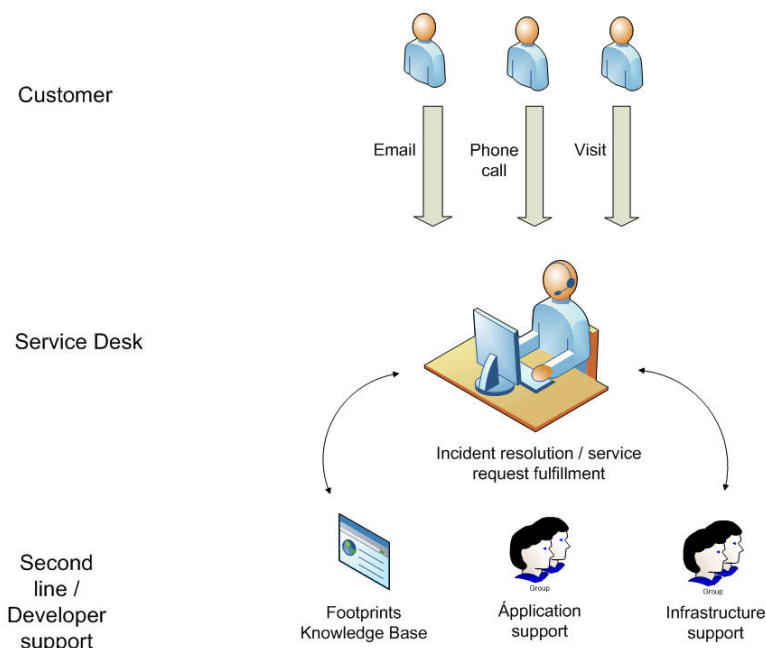
Kuvio 3. Tukikyselyn kysymykset 1-4, lähtötilanne 2010 - tulokset

2 Kehitystyön tavoite

Tämän opinnäytetyön tavoitteena on kehittää toiminnanohjausjärjestelmälle tehokkaampi ylläpitoprosessi, joka kattaa järjestelmätuen sekä järjestelmän pienkehitystehtävät. Prosessin kehittämisen yhteydessä tutustutaan ennalta valittuihin menetelmiin, joita ovat ITIL, Kanban sekä Scrum. Nämä menetelmät ovat maailmanlaajuisesti tunnettuja ja toimiviksi todettuja, joten tämän työn tarkoituksena on tutkia niiden soveltuvuutta yrityksen toiminnanohjausjärjestelmän tukeen ja pienkehitykseen. Nämä menetelmät on valittu tutkittavaksi, koska yrityksen emoyhtiö käyttää kyseisiä menetelmiä liiketoimintasovelluksiensa hallinnoimisessa sekä päivittäisten toimintojen ohjaamisessa. Näitä menetelmiä käsitellään luvusta 6 eteenpäin.

2.1 Palvelupisteen toiminnan parantaminen

Palvelun tuella viitataan tässä työssä palvelupisteen toimintaan, joka tarkemmin koskee toiminnanohjausjärjestelmän tukipyyntöjä. Tukiprosessissa on tarkoitus kuvata loppukäyttäjän näkökulmasta tukipyyntöjen kulku eri tukitasojen ja resurssien läpi sekä tuottaa palvelupisteelle tukipyyntöjen ratkaisuun kyselylomake sekä ohjeita järjestelmän teknisiin ongelmiin liittyen. Palvelupisteellä viitataan tässä työssä yrityksen sisäiseen palveluun, joka toimii yhteydenottokanavana erilaisille ongelmille sekä pyynnöille liiketoiminnan ja IT:n välillä. Yrityksen palvelupisteen rakenne on alla olevan kuvan mukainen (kuvio 4).



Kuvio 4. Palvelupisteen rakenne yrityksessä

Tukiprosessissa on tarkoitus käyttää resursseina palvelupisteen henkilöitä, järjestelmän avainkäyttäjiä, järjestelmän eri osa-alueista vastaavia henkilöitä eli prosessin omistajia sekä järjestelmän ylläpitoryhmään kuuluvia henkilöitä. Työn tavoitteena on myös kehittää palvelupisteelle alustava tukipyynnöjen käsittelyprosessi, joka pohjautuu ITIL:n teoriaan. Tukipyynnön käsittelyprosessissa kuvataan tukipyynnöjen kulku eri tukitasojen läpi, sekä oikeanlainen tukipyynnöjen kirjaaminen. Tukipyynnön käsittelyprosessissa ovat resursseina palvelupisteen henkilöt sekä ylläpitoryhmään kuuluvat henkilöt.

Loppukäyttäjälle suunnatussa prosessissa tärkeimpänä saavutuksena voitaisiin pitää sitä, että sen avulla saadaan käyttäjät ymmärtämään, miten tukiprosessi toimii tämän toiminnanohjausjärjestelmän osalta, jotta vähennettäisiin suoria yhteydenottoja yksittäisiin henkilöihin ja tätä kautta saataisiin resurssit keskittymään heille kuuluviin tehtäviin tukipyynnöjen selvittelyn sijaan. Palvelupisteelle kehitetyn tukipyynnön käsittelyprosessin tarkoituksena on kehittää käyttäjiltä tulevien tukipyynnöjen kirjaamista oikean tavan mukaisesti sekä nopeuttaa palvelupisteen toimia järjestelmän normaalitilan palauttamiseksi.

Edellä mainittujen prosessien lisäksi on tarkoitus tuottaa yleisimpiin tukipyyntöihin ohjeistukset, jotta palvelupiste pystyisi ratkaisemaan toistuvat järjestelmään liittyvät tukipyynnot itse ja välittämään ainoastaan vaativimmat, järjestelmän syvempää tietämystä vaativat tukipyynnot prosessissa eteenpäin oikealle resurssille. Nämä ohjeistukset eivät sisälly tähän versioon, koska ohjeistukset sisältävät yksityiskohtaista tietoa järjestelmästä.

Tämän työn ohessa on myös tarkoitus tehdä palvelupisteelle tähän toiminnanohjausjärjestelmään liittyen kyselylomake, josta selviää ongelman perustiedot. Kyselylomakkeen avulla saadaan tehostettua tukiprosessia sekä säästetään tukipyyntöä ratkaisevan henkilön aikaa, kun henkilöllä on pohjatiedot ongelmasta eikä hänen tällöin tarvitse käyttää enää aikaa ongelman perustietojen selvitykseen. Myös kyselylomakkeen avulla säästetään loppukäyttäjän aikaa, kun hänen ei tarvitse käyttää aikaa ongelman uudelleen selitykseen uudelle tukihenkilölle.

2.2 Pienkehityskäytäntöjen parantaminen

Pienkehityksellä tarkoitetaan järjestelmään tehtäviä toimenpiteitä, kuten järjestelmän ominaisuuksien korjaamista, parantamista tai uusien ominaisuuksien lisäämistä pienkehittämisen rajojen sisällä. Työn tavoitteena on tutkia Agile-menetelmistä Kanbania sekä Scrumia ja näiden soveltamista järjestelmän pienkehitysprosessiin. Näiden menetelmien tutkimisen jälkeen on tarkoitus etsiä näistä menetelmistä sopivia ominaisuuksia pienkehitysprosessiin tämän kyseessä olevan järjestelmän pienkehitykseen. Mikäli pienkehitykseen kehitetään prosessi, siinä on tarkoituksena käyttää resursseina järjestelmän ylläpitäjiä mukaan lukien yrityksen omat työntekijät sekä ulkopuoliset toimittajat. Pienkehitysprosessiin on myös tarkoitus ottaa mu-

kaan yrityksen liiketoiminnan puolelta järjestelmän prosessin omistajat. Pienkehitykseen liittyvän prosessin kehittämisen lisäksi tavoitteena on luoda kehitysideoita varten muutospyyntö-lomake, jonka avulla muutokset saadaan hallitusti ja järjestelmällisesti käsiteltyä.

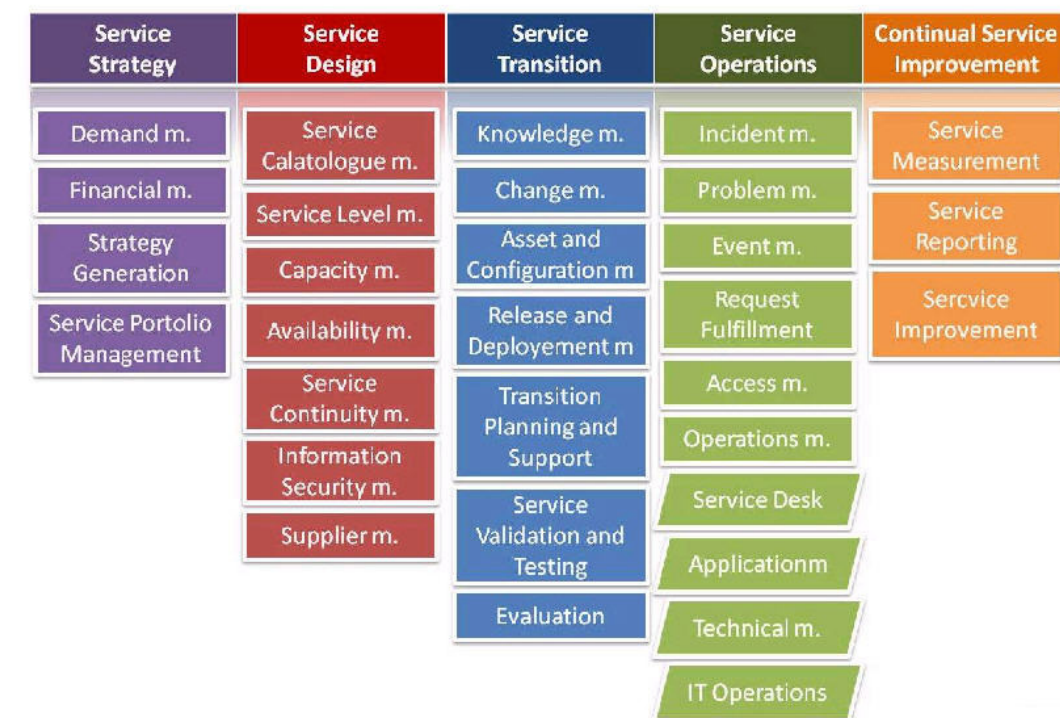
3 Keskeiset käsitteet

Seuraavaksi käydään lyhyesti läpi työtä koskevat käsitteet ja tämän jälkeen siirrytään työhön liittyvään teoriaan. Käsitteiden läpikäymisellä pyritään helpottamaan työn lukemista ja tekstissä käytettyjen termien ymmärtämistä.

3.1 Information Technology Infrastructure Library

Information Technology Infrastructure Library eli ITIL alun perin on kehitetty vuonna 1989 ja sen alkuperäinen tarkoitus oli ohjeistaa Englannin hallituksen toimia. ITIL:n perustaja oli CCTA:n (Central Computer and Telecommunication Agency) joka yhdistettiin myöhemmin osaksi Iso-Britannian valtiovarainministeriön virastoa OGC:a (Office of Government Commerce). OGC omistaa ITIL-kirjojen oikeudet ja on yhteistyössä muiden tahojen kanssa kehittänyt nämä parhaat käytännöt, joiden käyttö on laajennut ympäri maailmaa ja joita hyödynnetään maailmanlaajuisesti useassa yrityksessä palvelutoimintojen tehostamiseen sekä oikeanlaisten käytäntöjen löytämiseen. (Kupiainen 2007, 8.) ITIL:stä on kehitetty versio 3 vuonna 2007, joka jaottelee prosessit palvelun elinkaaren mukaan, joita ovat palvelustrategia, palvelun suunnittelu, palveluun siirtyminen, palvelun toimittaminen sekä palvelun jatkuva kehittäminen (ItSMF).

ITIL v3:n prosessit ja niiden aliprosessit on kuvattu selkeästi Petér Fehér kuvassa (kuvio 5). Versiossa kolme on otettu huomioon siis kaikki palvelun elinkaaren vaiheet, kun versiossa kaksi keskityttiin palvelun tukeen ja toimittamiseen eikä muita vaiheita oikeastaan käsitelty ollenkaan.



Kuvio 5. ITIL v3 rakenne (Fehér 2009).

3.2 Palvelupiste

Palvelupisteellä (Service Desk) viitataan yrityksen tukitoimintoon, joka vastaa asiakkaiden kyselyihin puhelimitse, sähköpostilla, sähköisellä järjestelmällä tai paikallisesti (ITIL 2007, 109). ”Palvelupiste käsittelee asiakkaiden ja käyttäjien yhteydenottojen kautta ilmoittamat tapahtumat (incident), ongelmatilanteet ja kysymykset. Lisäksi palvelupiste tarjoaa väylän muutospyyntöjen (request for change, RFC), ylläpitosopimusten ja ohjelmistolisenssiasioiden käsittelylle. Palvelupiste vähentää muiden tukiryhmien työmäärää vastaamalla kyselyihin, jotka voidaan selvittää ilman erityisasiantuntijoita” (Kupiainen 2007, 12-13). Palvelupiste voidaan toteuttaa kolmella eri tavalla jotka ovat paikallinen, virtuaalinen tai keskitetty palvelupiste. Paikallisessa palvelupisteessä tukitoiminnot on tuotu lähelle IT-palveluiden käyttäjiä ja palvelupiste onkin yleensä sijoitettu yrityksen tiloihin. Palvelupiste voidaan toteuttaa paikallisesti esimerkiksi seuraavista syistä, kuten kieli -ja kulttuurierot, fyysinen apu tai VIP-asiakkaat, jotka tarvitsevat esimerkiksi omalle työpisteelle jonkun auttamaan heitä. Virtuaalinen palvelupiste taas tarkoittaa palvelupistettä, joka on toteutettu sähköisesti eikä ole paikariippuvainen, jolloin tukihenkilöt voivat sijaita ympäri maailmaa. Virtuaalinen palvelupiste on toteutettu täysin sähköisesti käyttäen apuna yrityksen palvelupisteitä ja Internetiä. Keski-

tetyssä palvelupisteessä on kyse toiminnan tehostamisesta yhdistämällä useita palvelupisteitä. Keskitetyssä palvelupisteessä erilainen osaaminen yhdistyy ja tätä kautta toiminta tehostuu sekä kustannukset laskee. (ITIL 2007, 109-111.)

3.3 Palvelutasosopimus (SLA)

Palvelutasosopimus tunnetaan paremmin lyhenteellä SLA, joka tulee sanoista Service Level Agreement. SLA on asiakkaan ja palvelun tarjoajan välinen sopimus siitä, mikä on tarjottavan palvelun palvelutaso. SLA:n avulla pyritään siis takaamaan palvelun laatu asiakkaalle sekä sillä pyritään realisoimaan asiakkaan odotukset palvelusta. SLA:sta on hyötyä myös palveluntarjoajalle, koska sopimuksen avulla palvelun tarjoaja tietää paljon resursseja täytyy varata palvelutason säilyttämiseksi, kun palvelun odotukset ovat tiedossa. SLA voidaan tehdä kahden yrityksen välille tai yrityksen eri osastojen välisesti, esimerkiksi IT-osaston ja liiketoiminnan välille. SLA:ssa määritellään esimerkiksi tarjottava palvelu, vastuut, takuut sekä prioriteetit. SLA:ssa määritellään myös palveluntarjoajalle tulevat rangaistukset palvelutason alittamisesta sekä mittarit joilla palvelutasoa mitataan. (Kautto 2009, 1.)

3.4 Prosessi

Prosessi koostuu joukosta toisiinsa liittyviä toimintoja sekä niiden toteuttamiseen tarvittavia resursseja. Prosesseja voi olla erityyppisiä kuten liiketoimintaprosessi, ydinprosessi tai tukiprosessi. Liiketoimintaprosesseja ovat ne, jotka ovat kriittisiä yrityksen liiketoiminnan kannalta. Ydinprosessit ovat niitä, jotka on suunnattu suoraan ulkoiselle asiakkaalle. Tukiprosesseja ovat liiketoimintaa kannattelevat prosessit, kuten tietojärjestelmän käytön tuki ja kehitysprosessit. (Laamanen & Tinnilä 2009, 121-122.)

3.5 Tietojärjestelmä

Kun puhutaan tietojärjestelmästä, viitataan yleensä laitteista, ohjelmistoista, tietokannoista ja niiden välisistä vuorovaikutussuhteista koostuvaan järjestelmään, jonka tärkeimpiin tehtäviin kuuluu kaikenlainen informaation käsittely, kuten keruu, varastointi, analysointi sekä välitys. Myös järjestelmän käyttäjät voidaan nähdä osana tietojärjestelmää. (Koto 2009, 5.)

3.6 Toimintatutkimus

Toimintatutkimusta voidaan harjoittaa joko oman osaamisen kehittämiseen tai esimerkiksi jonkin yhteisön käytäntöjen parantamiseen. Parhaiten toimintatutkimus soveltuu ryhmän toiminnan muuttamiseen. Toimintatutkimuksen avulla pyritään siis ratkaisemaan erilaisia käytännön ongelmia. Tutkimuksen olennaisin piirre on muutos, joka on pysyvä ja saadaan aikaan vain toiminnan ja yhteisten tavoitteiden kautta. Toimintatutkimus ei ole pelkästään tutkijoiden työtä vaan siihen kuuluu olennaisena osana ihmiset joita muutos koskee, joten yhteistyö näiden ihmisten kanssa onkin varsin tärkeä osa tutkimustyötä. (Kananen, 2009. 9-13.)

Toimintatutkimuksen osa-alueisiin kuuluu toiminnan kehittäminen, yhteistoiminta sekä tutkimus. Ilman tutkimusta käytäntöjen parantaminen on vain arkipäivän parantamista. Toimintatutkimus eroaa muista perinteisistä tutkimusmenetelmistä siinä, että toimintatutkimuksessa tutkija on itse mukana tutkimuksessa kun taas perinteisissä tutkimusmenetelmissä tutkija ei saa vaikuttaa tutkimustuloksiin. (Kananen 2009, 13.)

Toimintatutkimuksen tiedonkeruumenetelmiin kuuluu mm. haastattelu, havainnointi, kirjalliset lähteet sekä kyselyt. Toimintatutkimus eroaa esimerkiksi laadullisesta tutkimuksesta siten, että laadullinen tutkimus käsittää ainoastaan suunnittelun, tiedonkeruun sekä analyysin kun toimintatutkimukseen kuuluu näiden lisäksi vielä toiminnat muutoksen tekoon. Eli voidaan sanoa, että toimintatutkimus alkaa siitä mihin laadullinen tutkimus loppuu. (Kananen 2009, 22.)

4 Ylläpito

Ylläpidolla tarkoitetaan yleisimmän määrittelyn mukaan yhtä vaihetta ohjelmistotuotannossa. Tarkemmin niitä toimintoja, joita ohjelmalle tehdään sen käyttöönoton jälkeen. Ylläpito alkaa siis, kun järjestelmä otetaan käyttöön ja ylläpito loppuu vasta kun järjestelmä poistetaan käytöstä. (Harsu 2003, 17-18.) Seuraavaksi käsitellään ylläpitoon ja sen hallintaan liittyviä asioita kuten mihin ylläpitoa tarvitaan, ylläpidon eri osa-alueet sekä ylläpitoon liittyvät ongelmat ja niihin kehitettyjä ratkaisuja.

4.1 Ylläpidon tarve

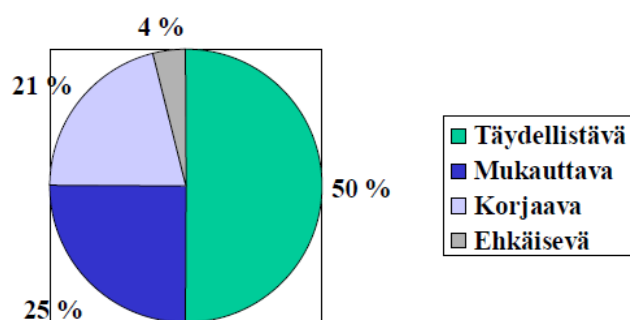
Järjestelmien ylläpidon tarve voi johtua useista eri syistä eikä se aina läheskään koske huonosti suunniteltua tai ohjelmoitua järjestelmää. Ylläpidolle voi olla tarvetta vaikka järjestelmä olisi koodattu parhaimpien käytäntöjen mukaisesti ja toteutettu yrityksen haluamilla komponenteilla ja toiminnallisuuksilla. Ylläpidon tarve tulee esille esimerkiksi silloin, kun

ohjelmat on suunniteltu käyttämään tiettyjä muisti- ja tilarajoituksia. Näiden takia järjestelmän komponentteja täytyy päivittää tilanteen mukaan sopivaksi, jotta ne sopivat yhteen uusien laitteiden ja käyttöjärjestelmien kanssa. Tämän seurauksena järjestelmän rakennetta on täytynyt myös muuttaa eli järjestelmälle on ollut tarvetta ylläpidollisiin muutoksiin. (Harsu 2003, 18; 79.) Ylläpitoa tarvitaan myös järjestelmän päivittämiseen muuttuneiden vaatimusten ja tarpeiden mukaisiksi sekä ilmenneiden virheiden korjaamiseksi. Muutostarpeita järjestelmälle voi luoda esimerkiksi yrityksen muuttuneet liiketoimintaprosessit taikka muuttuneet järjestelmää taikka sen tiettyä osaa ohjaavat lait. (Harsu 2003, 67.) Järjestelmään voidaan myös haluta lisää uusia toiminnallisuuksia liiketoiminnan tehostamisen kannalta tai työn helpottamisen kannalta, joten näiden lisääminen järjestelmään on myös osa ylläpitoa.

Harsun sekä Mäkelän mukaan ohjelmistoyrityksissä yli puolet työajasta kuuluu ylläpitotehtäviin ja Mäkelä mainitsee myös, että ylläpitotehtävissä eniten aikaa kuluu ohjelmiston toiminnan ymmärtämiseen ja jopa 90 % ylläpitoajasta voi olla tiedonhankintaa. Mäkelä on ottanut esiin seminaarissaan myös yhden mielenkiintoisen seikan koskien järjestelmän kustannuksia, jossa hän toteaa ylläpidon vievän 85 % järjestelmän kokonaiskustannuksista. (Harsu 2003, 18; Mäkelä 2004, 5.)

4.2 Ylläpidon osa-alueet

Ylläpito voidaan jakaa neljään osa-alueeseen, joita ovat korjaava, mukauttava, täydellistävä sekä ehkäisevä ylläpito. (Harsu 2010, 5) Tähän työhön liittyvät osa-alueet ovat korjaava ja täydellistävä ylläpito, mutta käydään nämä kaksi muuta osa-aluetta myös läpi lyhyesti. Kuviosta 6 näkee selvästi, miten ylläpitotyöt jakautuvat Harsun mukaan eri ylläpitotyyppien kesken.



Kuvio 6. Ylläpidon osa-alueet jaoteltuna (Harsu 2010, 2)

Korjaavaa ylläpitoa tarvitaan silloin, kun järjestelmä ei toimi niin kuin sen vaatimusmääritellyn mukaan pitäisi toimia. Korjaavat toimenpiteet voidaan luokitella kiireellisiin ja vuorollaan tehtäviin korjauksiin. Kiireellisiin korjauksiin lasketaan sellaiset virheet, jotka aiheuttavat

esimerkiksi yrityksen liiketoiminnalle haittaa, taikka jolla on vaikutusta usean henkilön työtehtäviin siten, etteivät henkilöt pysty ennen virheen korjaamista jatkamaan työskentelyä. Vuorollaan tehtäviin korjauksiin voidaan lukea virheet, joilla taas ei ole vaikutusta liiketoimintaan eikä virhe vaikuta usean henkilön työskentelyyn. (Harsu 2003, 86-87.)

Mukauttavaan ylläpitoon Harsu luokittelee järjestelmän muuttamisen laitteiden, siihen liittyvien ohjelmien sekä käyttöjärjestelmien uusiutumisen seurauksena. Tähän kategoriaan kuuluu esimerkiksi muistin, levytilan tai tietoliikenneopeuksien lisääminen, jotta järjestelmä toimii sille asetettujen vaatimusten mukaisesti. (Harsu 2003, 88.)

Täydellistävään ylläpitoon kuuluu uuden toiminnallisuuden lisääminen, joka voi olla esimerkiksi yrityksen liiketoiminnan kilpailukyvyyn kannalta tärkeä. Uuden toiminnallisuuden lisääminen voi tapahtua esimerkiksi muutospyyntödokumentin perusteella, jonka muutosta haluava henkilö täyttää. Muutospyyntödokumentin avulla pystytään kommunikoimaan paremmin IT-asiantuntijoiden ja käyttäjien välillä, koska sen avulla saadaan selville tarkemmin käyttäjien tarpeet, toiminnallisuuden kuvaus sekä saadaan välitettyä tarkemmat muutoksen tiedot sitä tarvitseville henkilöille kuten ohjelmoijille, suunnittelijoille ja testaajille. (Harsu 2003, 82.) Ennen uuden toiminnallisuuden lisäämistä on selvítettävä muutoksen vaikutus koko järjestelmään esimerkiksi järjestelmän suunnitteludokumentin, kuten tietovuokaavion avulla. Myös muutoksen vaikutukset tietokantoihin, tiedostoihin ja tietorakenteisiin on hyvä selvittää ennen uuden toiminnallisuuden tekemistä. Kun on selvitetty muutettavat ohjelmat ja moduulit uuden toiminnallisuuden lisäämiseksi, tarkastetaan näiden dokumentaatiosta muutostarpeet ja päivitetään nämä dokumentit. (Harsu 2003, 88-89.)

4.3 Ylläpitoon liittyvät ongelmat

Järjestelmän ylläpitoon liittyy myös varjopuolia, kuten koodin huono rakenne, dokumentaation puuttuminen tai vajavaisuus, ylläpitäjän puutteelliset tiedot järjestelmästä, ylläpitotehtävien huono maine ja uusien virheiden esiintyminen. (Harsu 2003, 78.)

Huonolla koodin rakenteella viitataan siihen, kun järjestelmiä muutetaan tarpeiden muuttuessa, niin koodin rakenne muuttuu monimutkaisemmaksi. Esimerkiksi muutos on toteutettu järjestelmään siten, että koodiin on lisätty pitkiä aliohjelmia ja monimutkaisia moduuleita, jolloin koodin rakenne on vaikeaselkoista. (Harsu 2003, 78.)

Koodin huono rakenne johtaa taas ylläpitäjän riittämättömiin tietoihin järjestelmästä ja sovellusalueesta, kun muutoksia ei ole dokumentoitu ja koodin lukeminen on työlästä. Ylläpitäjän täytyy tietää järjestelmän rakenne jotta pystyy ylläpitämään sitä. Usein ylläpitäjät joutuvatkin hankkimaan järjestelmän tiedot koodista, koska eivät voi luottaa dokumentaation

paikkansapitävyyteen. Järjestelmätietoja saattaa joutua etsimään myös useasta eri lähteestä, joita voi olla esimerkiksi määrittely- ja suunnitteludokumentit, käyttöohjeet, konfiguraatiodat, järjestelmä itsessään sekä järjestelmän käyttäjät. Mäkelä kiteyttää järjestelmätietämyksen tärkeyden hyvin lauseessaan ”Jotta järjestelmälle voidaan suorittaa ylläpitoa, on saatava pohjatietoa itse järjestelmästä.” (Mäkelä 2004, 5.) Eli tämä tarkoittaa sitä, että ylläpitäjän täytyy saada perustiedot järjestelmästä, rakenteesta sekä toiminnoista, jotta voi ylläpitää järjestelmää, eikä pelkkä järjestelmän rakenteen tuntemus riitä kuten Harsu kirjoitti. (Harsu 2003, 78; Mäkelä 2004, 6.)

Dokumentaation puuttuminen tai vajavaisuus vaikeuttaa huomattavasti ylläpitotehtäviä. Syynä dokumentaation puuttumiseen sekä vajavaisuuteen on yleensä kiireellinen aikataulu sekä dokumentaation kirjoittamisen priorisointi. Muutosten seurauksena järjestelmän rakenteesta on voinut tulla vaikeaselkoinen ja epälooginen ja usein järjestelmämuutoksia tehtäessä dokumentointi jää toissijaiseksi tehtäväksi. (Harsu 2003, 78-79.)

Yhtenä syynä ylläpidon ongelmiin on myös tehtävien mielekkyys. Ylläpidossa joutuu olemaan yhteydessä asiakkaisiin eikä yhteydenpito loppukäyttäjän puolelta ole aina niin positiivista. Myös koodin huono laatu vaikuttaa ylläpitotehtävien mielekkyyteen negatiivisesti. (Harsu 2003, 79.)

Ylläpitotehtäviin valitaan myös yleensä kokemattomia ohjelmoijia vaikka niihin pitäisi valita kokeneita ohjelmoijia, koska koodin tulkitseminen on haastavaa jo kokeneemmallekin ohjelmoijalle. (Harsu 2003, 79.) Harsu viittaa ohjelmien ylläpito ja evoluutio-luennossa Robert L. Glassin tutkimukseen, jossa hän on tutkinut ylläpitotehtävien mielekkyyttä, osoittaakseen ylläpitotehtävien suosion. Tutkimuksessa ylläpitotehtävät arvosteltiin kaikista tylsimmiksi kyselyssä mukana olleista tehtävistä. (Harsu 2010, 1.) Ylläpitotehtävien mielekkyydestä johtuen ylläpidossa olevien henkilöiden vaihtuvuus voi olla kohtalaisen suuri, jolla voi olla suuriakin vaikutuksia järjestelmän kokonaiskustannuksissa, koska aiemmin tässä työssä on jo mainittu, että tiedon hankkimiseen menee ylläpidossa eniten aikaa..

Ylläpidossa on otettava myös huomioon vaikutukset, jotka ovat seurausta virheiden korjaamisesta tai uusien ominaisuuksien lisäämisestä. On tärkeää testata virheiden korjaukset sekä uudet ominaisuudet huolella ettei niistä aiheudu uusia virheitä (väreilyvaikutus). Väreilyvaikutuksella tarkoitetaan uusien virheiden korjaamisen kautta tulevia uusia virheitä. Yleensä virheiden syntymistä ei pystytä millään täysin estämään, koska testaaminen ei yleensä kata kaikkia mahdollisia järjestelmän osa-alueita. Väreilyvaikutuksia voi olla siis hyvin vaikea välttää, koska jopa yksinkertaisimmatkin korjaukset voivat aiheuttaa järjestelmän johonkin toiseen osaan toimintahäiriöitä, jotka tulevat ilmi vasta järjestelmän käytössä. (Harsu 2003, 77.)

Väreilyvaikutuksen seurauksena ylläpidon työtaakka saattaa vain kasvaa, mikäli järjestelmän testausta ei pystytä suorittamaan jokaisen käyttäjän näkökulmasta.

Ylläpitoon liittyviin ongelmiin on myös mietitty ratkaisuja. Mäkelän mukaan ylläpitoon kulu-
tettua aikaa pystytään vähentämään tuntemalla järjestelmä paremmin. Hän sanoo, että par-
haiten järjestelmän oppii, jos on hyvä ohjelmointikielten ymmärrys ja kokemus, sekä sen oh-
jelmointikielen tuntemus jolla kyseinen järjestelmä on toteutettu. (Mäkelä 2004, 5.) Eli tämä
voitaisiin ymmärtää niin, että ylläpitotehtäviin pitäisi ottaa vain kokeneita ohjelmoijia, koska
aloittelevan ohjelmoijan on huomattavasti vaikeampi lukea toisen kirjoittamaa koodia kuin
itse kirjoittaa sitä. Jos tuntemalla järjestelmän hyvin pienennetään ylläpitokustannuksia,
pitäisi siis saada vaihtuvuus ylläpidossa pienemmäksi esimerkiksi tehtävien mielekkyyden ja
haastavuuden nostamisella. Harsu on koonnut kirjassaan ylläpitokustannuksiin liittyviä ratkai-
suja, joita ovat koodin rakenteen ja laadun parantaminen, standardien noudattaminen ohjel-
moinnissa ja dokumentoinnissa, käyttäjien mukaan ottaminen suunnitteluvaiheessa sekä tule-
vien muutoksien ennakoiminen. (Harsu 2003, 81.) Harsun listaamat ratkaisutkaan ei aivan
helposti ole toteutettavissa. Esimerkiksi käytössä olevan järjestelmän koodin rakenteen laa-
dun parantaminen ei välttämättä käy ihan kädenkäänteessä, mutta luomalla ylläpitoon stan-
dardeja, kuten miten täytyy dokumentoida muutokset tai millä ohjelmointikielellä järjestel-
mää ohjelmoidaan, sekä näiden standardien noudattamista valvomalla mahdollisesti pysty-
tään hallitsemaan ylläpidon kustannuksia ja mahdollisesti jopa laskemaan niitä.

4.4 Muutosten hallinta

Järjestelmään tehtävät ylläpidolliset toimet liittyvät läheisesti muutoksen hallintaan, koska muutosten tekeminen järjestelmään on ylläpitoa ja ylläpito on virheiden korjaamista ja toiminnallisuuden lisäämistä. Muutoksien esiintuomiseksi voidaan käyttää jo aiemmin mainittua muutospyyntödokumenttia. Tämän dokumentin avulla saadaan selville asiakkaan tarpeet sekä muutoksen vaikutukset järjestelmään. Dokumentista tulisi käydä ilmi asiakkaan toimesta ainakin tunnistenumero, muutoksen pyytäjä, muutoksen tyyppi (korjaava/uusi toiminnallisuus), havaintopäivämäärä (milloin virhe huomattu) sekä tavoitepäivämäärä (jolloin virhe korjattu), järjestelmä jota muutos koskee, virheen kuvaus ja muutoksen kuvaus. (Harsu 2003, 82.) Ohessa on yksi esimerkki muutospyyntödokumentista.

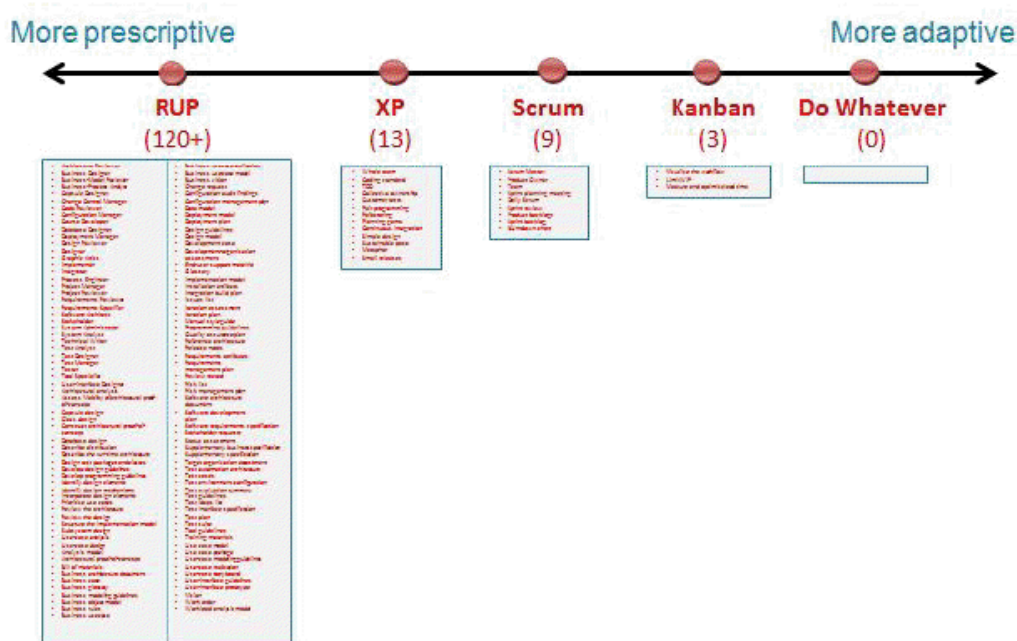
Tunnistusnumero	8603-002
Muutoksen pyytäjä	L. J. Green, NASA
Havaintopäivämäärä	04.03.86
Tavoitepäivämäärä	04.03.86
Ylläpitotyyppi	Korjaava
Muutoksen vakavuus	1
Järjestelmä	Titan Launch
Ohjelma	Ascent
Muutoksen kuvaus	Satelliitin palauttaminen määrätylle kiertoradalleen. Virheellisen nousuradan vuoksi satelliitti joutuu väärälle kiertoradalle.
Muutoksen hyödyt	Satelliitin ohjaaminen oikealle kiertoradalle maksaisi 80 miljoonaa dollaria, joten laukaisuohjelmaa ei voida jatkaa ilman korjausta.
Ongelman alkuperä:	
Järjestelmä	Titan
Ohjelma	Ascent
Moduuli	Initialization
Virheen tyyppi	Virheelliset alkuarvot
Ratkaisu	Alkuarvojen korjaaminen
Vaikutukset:	
Järjestelmät	Titan
Ohjelmat	Ascent
Moduulit	Initialization
Dokumentointi	Initialization-moduulin määrittelydokumentti
Vaihe	Ylläpito
Aloituspäivämäärä	05.03.86
Hyväksyntä	Tom Prieve
Henkilöt	Jay Arthur
Arvioidut resurssit	2 henkilötyöpäivää
Todelliset resurssit	1.5 henkilötyöpäivää

Kuvio 7. Esimerkki muutospyyntödokumentista (Harsu 2003, 84).

5 Kanban

Kanban syntyi vuonna 2004, kun David Andersson loi uudenlaisen ajattelutavan järjestelmäkehityksen kilpailusta ja rajoituksista asiantuntija Don Reinertsenin ohjauksella (Crisp AB). Tämä menetelmä kuuluu ketteriin ohjelmistokehitysmenetelmiin eli Agile-menetelmiin. Agile

sisältää käsitteitä sekä periaatteita siitä, miten järjestelmiä tulisi kehittää. Näitä periaatteita on lueteltu Pasi Reinin opinnäytetyössä, joita ovat esimerkiksi yksilöiden ja vuorovaikutustaitojen arvostaminen prosessien ja työkalujen sijaan, toimivan ohjelman arvostaminen dokumentaation sijaan sekä yhteistyön arvostaminen asiakkaan kanssa hankalien sopimusneuvotteluiden sijaan. (Reini 2008, 24-25.) Agilen menetelmissä pyritään käyttämään resurssit mahdollisimman tehokkaasti yhteistyössä asiakkaan kanssa saamalla aikaan paras mahdollinen lopputulos ja mahdollisimman lyhyessä ajassa. Agile-menetelmiä hyödykseen käyttävät työskentelevät pienissä, asiantuntevissa ja itseohjautuvissa tiimeissä, joille annetaan valtaa ja vastuuta tekemisestään (Lukumaa 2006.) Agile-menetelmät ovat siis joustavia ja antavat menetelmää noudattaville vapauden toteutukseen ainoana vaatimuksena tavoitteen saavuttaminen määritetyssä ajassa. Oheisessa kuvassa verrataan eri menetelmiä, joita käytetään järjestelmän kehityksessä ja ylläpidossa ja jossa on kuvattu jokaisen eri menetelmän rajoitukset ja luokiteltu menetelmät joko määrääväksi tai joustavaksi. Kuten kuviosta 8 voi huomata, niin Kanban on näistä kaikista menetelmistä joustavin ja eniten sovellettavissa olevin menetelmä.



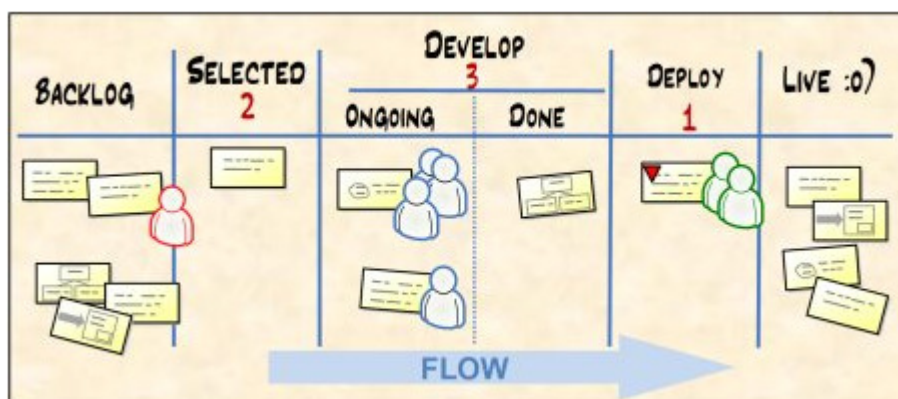
Kuvio 8. Eri menetelmien joustavuuden vertailu (Kniberg 2009, 9).

5.1 Kanbanin periaatteet

Kanban-menetelmä perustuu kolmeen asiaan, joita ovat työmäärän visualisointi, työmäärän rajoitus sekä saavutetun ajan mittaaminen. Työmäärän visualisoimisella tarkoitetaan sitä, että tehtävät kirjoitetaan esimerkiksi paperilapuille ja nämä laput kiinnitetään taululle jokainen omaan sarakkeeseen, josta nähdään missä vaiheessa mikäkin tehtävä on menossa. Työmäärän rajoittamisella tarkoitetaan sitä, että jokaiseen sarakkeeseen määritellään maksimi tehtävämäärä, eli kuinka monta tehtävää saa maksimissaan kyseisessä sarakkeessa olla sa-

maan aikaan. Kun työn alla olevien määrä on rajoitettu, huomataan ohjelmistokehityksessä tulevat virheet nopeammin (Crisp AB). Saavutettu aika saadaan mittaamalla yhteen tehtävään kulunut keskivertoaika (kierrosaika). Kanbanissa ei siis määritellä kaikkien tehtävien yhteis-kestoja vaan yhden tehtävän keskivertoaika. (Kniberg 2009, 6.)

Kuviossa 9 on esimerkki Kanbanin työpöydästä, jossa on tehtävälistalla olevat tehtävät, valitut tehtävät, työn alla olevat tehtävät, testissä olevat tehtävät sekä valmiit tehtävät. Punaisella merkityt numerot viittaavat siis siihen, että sarakkeessa ei saa olla enempää tehtäviä kuin mikä numero tuohon tilaan on määritelty. (Kniberg 2009, 14.)



Kuvio 9. Kanbanin työpöytä (Kniberg, 2009, 1).

5.2 Kanbanin määrittelemä työskentelymalli

Kanbanin ohjelmiston kehitysprosessi on visualisoitu niin, että kukin työvaihe luodaan omaksi sarakkeeksi niin kuin kuviossa 9 oli tehty, mutta olisi kuitenkin parempi, jos jokaisessa vaiheessa olisi myös valmis-vaihe, jossa ideaalitilanteen mukaan olisi aina vähintään yksi tehtävä mutta korkeintaan kolme, jottei seuraava työvaihe joudu odottamaan uutta työtä. Valmis-sarakkeen avulla tehtävän läpimenoaika saadaan mahdollisimman lyhyeksi, koska odotusaikaa ei tehtävän kierrossa ole. (Pirinen 2010, 43). Tiimin jäsenien vastuulla on arvioida tehtävään kuluva aika ja asiakkaan vastuulla taas on laittaa tehtävät tärkeysjärjestykseen. Mikäli tiimin jäsen kokee osaavansa tehdä myös seuraavan työvaiheen, hän voi suorittaa sen ja siirtää siten tehtävän seuraavaan sarakkeeseen. (Pirinen 2010, 42.)

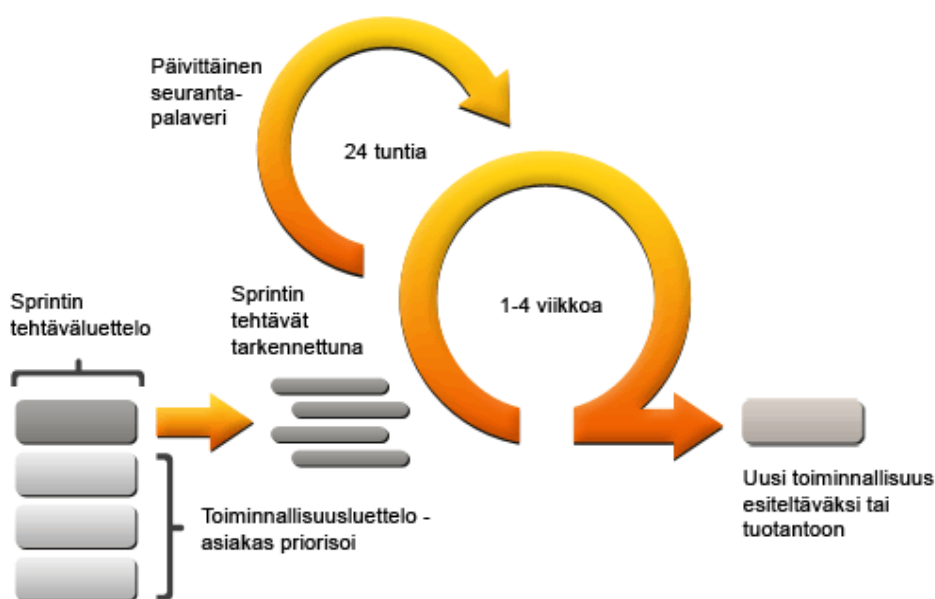
6 Scrum

Scrum-prosessi kuuluu myös Agile-menetelmiin ja se on yksi käytetyimmistä Agile-menetelmistä. Ensimmäinen Scrum-projekti syntyi vuonna 1993 sen kehittäjien Jeff Sutherlandin, John Scumniotalesin ja Jeff McKennan toimesta. Scrum-menetelmä julkistettiin ensimmäistä kertaa vuonna 1996. Menetelmä on kehitetty Harvard Business Review -artikkelin

“The New Product Development Game” (1986) - innovaatioista, jossa esiteltiin erityisen tuotavien yritysten tapoja tehdä tuotekehitystä. (Lindström 2006.)

Scrum-menetelmä tarjoaa ohjelmistokehitykseen mallin, jonka avulla ohjelmistokehitystä ohjataan. Scrumin avulla pystytään kontrolloimaan projektin etenemistä sekä sen avulla ohjelmistokehitystä pystytään vaiheistamaan eripituisiin jaksoihin. Scrumissa ohjelmistokehitystä tehdään sprinteissä, joiden sisällä tiimi synkronoi omia tehtäviään päivittäispalaverien avulla. Sprintillä tarkoitetaan järjestelmän kehityksessä yhtä jaksoa, joka on yleensä viikosta neljään viikkoon riippuen yrityksen määrittelemästä ajasta, jonka jälkeen järjestelmän uuden version pitäisi olla esitys/julkaisukelpoinen. Päiväpalaverilla tarkoitetaan 5-15 minuutin pituista päivittäistä tilannekatsausta tiimin kesken. (Koskela 2007, 2.)

Kun tarkastellaan hieman kuviota 10, niin kaikki alkaa siitä, kun asiakas määrittelee tiimille toiminnallisuusluettelon (Product Backlog) eli listan halutuista ominaisuuksista. Työn onnistumisen kannalta tämä on tärkein dokumentti tiimille, koska tämän perusteella tiimi työskentelee. Tämän jälkeen jokaisen sprintin alussa tiimi yhdessä asiakkaan kanssa valitsee sprintin sisällön, josta sitten muodostuu tiimille tehtäväluettelo (Sprint Backlog). Sprintin aikana tiimi työskentelee parhaiten katsomallaan tavalla, jotta jäsenet saavat toteutettua kaikki tehtäväluettelossa määritellyt ominaisuudet. Tiimi kokoontuu päivittäin 5-15 minuutin tapaamiseen keskustellakseen työn etenemisestä ja esiin tulleista ongelmista, jotta ongelmiin voidaan tarttua välittömästi ja tehtävälistan työt valmistuvat ajallaan.. (Koskela 2007, 2; Lindström 2006.)



Kuvio 10. Scrum-prosessi (Lindström 2006, 2).

6.1 Scrumin määrittelemät roolit

Scrum-menetelmään kuuluu kolme eri roolia, jotka ovat tuotteen omistaja, tiimi ja Scrum Master. Roolien sisään voi lukeutua useita eri henkilöitä eri osaamisaloilta.

Tuotteen omistajalla tarkoitetaan henkilöä, joka edustaa kehitettävän järjestelmän käyttäjiä, on vastuussa kustannuksista ja liiketoiminnan kannalta oikeiden asioiden kehittämisestä.

Tuotteen omistaja määrää myös, mitä ominaisuuksia milloinkin kehitetään. Tähän rooliin voi sisältyä useampi henkilö, jotka sitten yhdessä muodostaa halutun tahtotilan esimerkiksi järjestelmälle eli päättävät mitä asioita tehdään ja mitä ei. (Koskela 2007, 5-6.)

Tiimi koostuu eri osa-alueiden osaajista, kuten esimerkiksi ohjelmoinnin, järjestelmätestauksen ja teknisen dokumentoinnin ammattilaisista. Tiimin tehtävänä on siis käytännössä luoda tehtävälistan ominaisuudet toimiviksi ominaisuuksiksi. Tiimin sisällä henkilöillä ei ole rooleja, eikä tehtäviä myöskään määrätä vaan jokainen henkilö voi valita tehtävälistalta tehtäviä itselleen. Yleensä henkilöt valitsevat tehtävät oman osaamisalueen mukaan mutta tällainen käytäntö myös edesauttaa henkilökohtaisen osaamisalueen laajentamista. Kukaan ei myöskään määrää käytettäviä työkaluja tai menetelmiä vaan tiimissä luotetaan siihen, että jokainen henkilö tietää itselleen parhaiten sopivan ratkaisun tehtävien toteuttamiseksi. (Koskela 2007, 4-5.)

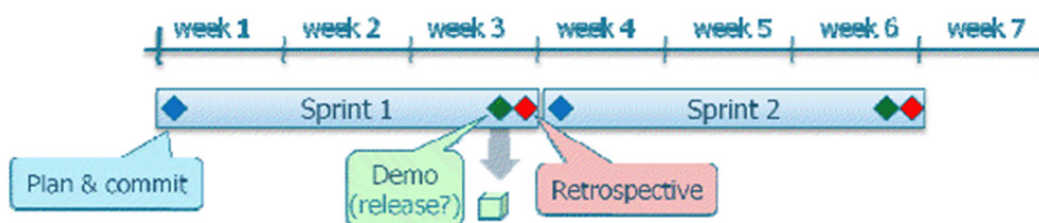
Scrum Master hallitsee tätä koko prosessia, ja roolille on myös määritelty viisi tärkeintä tehtävää. Nämä viisi tärkeintä tehtävää ovat Koskelan listauksen mukaan 1) esteiden poistaminen IT:n ja asiakkaan välillä siten, asiakas ohjaa kehitystä suoraan ilman välikäsiä, 2) Scrum työskentelytapojen opettaminen asiakkaalle, 3) tiimin olosuhteiden parantaminen tukemalla ja kannustamalla luovuutta ja itseohjautuvuutta, 4) tiimin tuottavuuden kasvattaminen sekä 5) menetelmien ja käytäntöjen parantaminen siten, että jokaisen Sprintin lopputulos on teknisesti tuotantokelpoinen. (Koskela, 5.) Eli Scrum Masterin tehtävänä on organisoida prosessin eteneminen, poistaa häiriötekijöitä sekä taata sujuva yhteistyö tiimin ja asiakkaan kanssa. Koskela onkin maininnut tekstissään, että hyvän Scrum Masterin täytyy omata paksu nahka, ihmistuntemusta sekä poliittista pelisilmää saadakseen Scrum-prosessin toimimaan. Scrum Master ei ole pysyvä rooli, vaan sitä voi kierrättää tiimissä, esimerkiksi silloin jos jäsenet eivät ole kiinnostuneita ottamaan roolia vastaan, jolloin tiimin jäsenet voivat olla sprintin ajan Scrum Masterin roolissa, minkä jälkeen rooli siirtyy seuraavalle jäsenelle. (Koskela 2007, 5-6.)

6.2 Scrumin mukainen työskentely

Scrumissa käytetään hyvin paljon samaa työskentelytapaa kuin Kanbanissa, eli tehtävät sijoitetaan tehtävätaululle, johon on määritelty eri vaiheet sarakkeisiin. Scrumia noudattaessa

yhden Sprintin aikana suoritetaan ainoastaan sovitut tehtävät eikä tehtävälistalle pysty lisäämään uusia tehtäviä.

Kuviossa 11 on esitetty Scrumin työskentelyvaiheet. Scrumissa on siis kolme eri aktiviteettia yhden Sprintin (kehitysjakson) sisällä. Ensimmäisessä vaiheessa (Plan & Commit) tiimi on valinnut tehtävälistalle tuotteen omistajan määrittelemät tehtävät prioriteetin mukaan sekä sen mukaan, mitä tiimi uskoo pystyvänsä tekemään sprintin aikana. Tämän jälkeen nämä valitut tehtävät suunnitellaan ja niiden tekeminen aloitetaan. Seuraavassa vaiheessa (Demo) tehtävät testataan ja niiden pitäisi olla valmiina julkaisuun. Kolmannessa vaiheessa (Retrospective) tiimi keskustelee sprintin aikana käytetyistä työskentelytavoista ja miettii miten niitä olisi voitu parantaa. (Kniberg 2009. 12-13.)



Kuvio 11. Scrum työvaiheet (Kniberg 2009, 12).

7 Palvelun toimittaminen

Palvelun toimittaminen on osa ITIL v3:sta. ItSMF (Information technology service management) on määritellyt palvelun tuottamisen seuraavasti: Palveluiden tuottaminen tarkoittaa palvelun toimittamista sovittujen tavoitteiden sekä käyttäjien ja asiakkaiden näkökulmasta. Palveluiden tuottaminen näkyy päivittäisessä työssä sekä tasapainon löytämisessä luotettavuuden ja kustannusten välillä. Palvelun toimittamiseen kuuluu myös herätteiden (event) hallinta, tapahtuman (insidenttien) hallinta, ongelmanhallinta, palvelupyyntöjen toteuttaminen, omaisuudenhallinta, palvelupiste (Service Desk), tekninen ja sovellushallinta sekä palvelutuotantoon liittyvän henkilöstön roolit ja vastuut. (ItSMF.) Tässä työssä käsitellään edellä mainituista prosesseista tapahtumanhallinta sekä palvelupyyntöjen toteuttaminen.

7.1 Tapahtumanhallinta

Tapahtumalla (Incident) tarkoitetaan normaalista käytöstä poikkeavaa IT-palvelun tapahtumaa kuten suunnittelematonta järjestelmäkatkosta tai järjestelmän laadun alenemista. Laadun aleneminen voi olla seurausta esimerkiksi tietoliikenneyhteyksien katkeamisesta tai levytilan loppumisesta. Tapahtumanhallinta on taas prosessi, joka käsittelee kaikki nämä tapah-

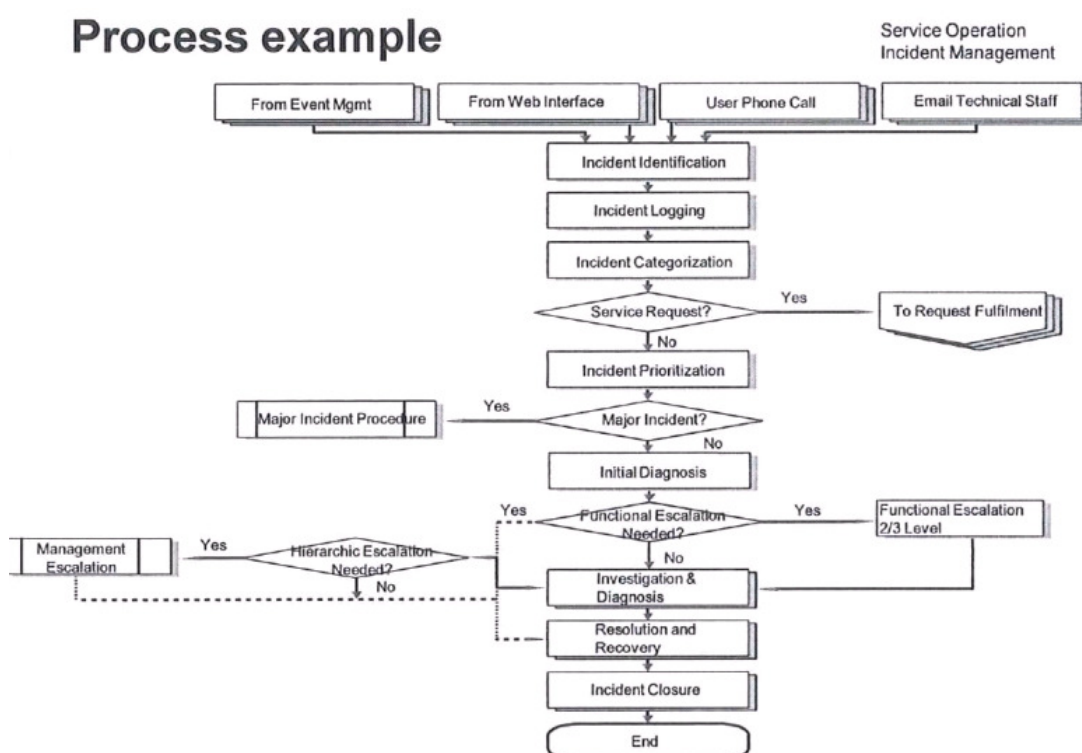
tumat kuten esimerkiksi järjestelmäviat, kysymykset tai tiedustelut. Edellä mainitut tapahtumat ovat yleensä käyttäjien ilmoittamia tapahtumia palvelupisteeseen, mutta ne voivat olla myös automaattisten hälytysjärjestelmien tai teknisten henkilöiden ilmoittamia. Tapahtumaksi ei siis lasketa käyttäjiltä tulleita palvelupyyntöjä, koska niissä ei ole kyse palvelun keskeytyksestä vaan yleensä käyttäjän tarpeiden täyttämisestä. (ITIL 2007, 46-47.) Tapahtumasta voidaan myöhemmin luoda ongelma, joka siirretään ongelmanhallintaan ratkaistavaksi. (Nylander & Kirkkala 2008, 4.)

Tapahtumanhallinnan tärkein tehtävä on palauttaa IT-palvelu mahdollisimman nopeasti normaalitasolle sekä minimoida liiketoimintaan ohjautuvat haitalliset vaikutukset samalla ylläpitäen hyvää palvelutasoa ja palvelun saatavuutta. Tapahtumanhallinta sisältää siis kaikkien mahdollisten tapahtumien käsittelyn, jotka keskeyttävät tai voivat keskeyttää IT-palvelun. Tapahtumanhallinta tuottaa arvoa liiketoiminnalle esimerkiksi pidentämällä IT-palvelun käyttöaikaa havaitsemalla ja ratkaisemalla tapahtumat ennakoidusti sekä havaitsemalla IT-palvelun kehittämistarpeet selvittämällä tapahtuman muodostumisen aiheuttajat. Tapahtumanhallinnassa on otettava huomioon tapahtumien käsittely- ja ratkaisuaajat sekä erilaiset tapahtumamallit. Käsittely- ja ratkaisuajoilla tarkoitetaan sitä, kuinka nopeasti tapahtumaan on vastattava tai kuinka nopeasti tapahtuma on ratkaistava. Käsittely ja ratkaisuaajan tulisi perustua SLA:ssa (Service Level Agreement), OLA:ssa (Operational Level Agreement) sekä UC:ssa (Underpinning Contract) sovittuihin aikoihin ja nämä ajat tulisi saattaa jokaisen osapuolen tietoon. (ITIL 2007, 46-47.) OLA-sopimus tehdään yrityksen sisäisesti esimerkiksi eri IT-toimintojen välille ja UC-sopimus tehdään toimittajan ja yrityksen välille (Certification.info 2009).

Tapahtumamallilla tarkoitetaan tunnetun tapahtuman ratkaisumallia, jota voidaan soveltaa tapahtumaan, joita esiintyy ajoittain ja johon ei ole pysyvää ratkaisua. Tapahtumamallin avulla pystytään siis varmistamaan yleisten tapahtumien käsittely oikealla, ennalta määritellyllä tavalla. Merkittävillä tapahtumilla (major incident) pitäisi tehdä oma tapahtumamalli, jotta varmistetaan tällaisten tapahtumien käsittely oikeassa ajassa ja oikealla prioriteetilla. Tapahtumamallit tulisi määrittelyjen jälkeen viedä tapahtumanhallintajärjestelmään ja automatisoida niiden käyttö. (ITIL 2007, 46-47.) Kuviossa 12 on kuvattu ITIL:n mukainen tapahtumanhallintaprosessin kulku.

Tapahtumanhallinta liittyy useaan eri ITIL:n määrittelemään toimintoon, jotka ovat ongelmanhallinta, asetushallinta, muutostenhallinta, kapasiteetinhallinta, saatavuuden hallinta sekä Service Level Management (SLM). Käydään lyhyesti läpi miten tapahtumanhallinta liittyy näihin. Tapahtuma on yleensä seurausta piilevästä ongelmasta, johon tarvitaan ongelmanhallinnan apua selvittämään ongelma, jottei tapahtumia enää ilmene. Asetushallinnan avulla pystytään selvittämään vialliset laitteet sekä arvioimaan tapahtuman laajuus. Esimerkiksi jos

yksi tietoliikennereititin olisi viallinen, niin asetushallinnan kautta nähdään vaikuttaako tämä ongelma kaikkiin järjestelmän käyttäjiin vai vain osaan niistä. Muutoksenhallinnan kautta hyväksytetään tarvittavat toimenpiteet, jotka vaativat muutosta järjestelmässä sekä hyväksytetään myös väliaikaiset ratkaisut (workaround). Kapasiteetinhallinta pystyy kehittämään väliaikaisia ratkaisuja tapahtumien selvittämiseksi, mikäli tapahtumanhallinnan suorituskyky laskee liikaa. Saatavuudenhallinta käyttää tapahtumanhallinnan tietoja selvittääkseen IT-palveluiden saatavuuden sekä määrittääkseen mitä tapahtuman elinkaaren vaihetta voidaan parantaa. SLM:n tehtävänä on määritellä palvelutasot, joita tapahtumanhallinta noudattaa, kuten ratkaisuaika, vaikuttavuusmäärittelyt, tavoitekorjausajat, palvelun määrittely asiakkaalle sekä säännöt palveluiden saamiseksi. (ITIL 2007, 53-54.)



Kuvio 12. Tapahtumanhallintaprosessin kulku ITIL:n mukaan

Tapahtumanhallinta tarvitsee myös tietoa tapahtumien ratkaisemiseen. ITIL:n mukaan tapahtumanhallinnalla tulisi olla käytössä jonkinlainen tapahtumanhallintatyökalu, joka sisältää tietoa erilaisten tapahtumien ja ongelmien historiasta, kategorioista, erilaisista ratkaisuista sekä erilaisia vianmäärittyskriptejä tapahtumien selvittämiseksi. Käytettävissä pitäisi olla myös tietokanta tunnetuista ongelmista sekä asetushallintajärjestelmä, jota käytetään IT-palveluntarjoajien asetusten määrittelyyn sekä voi myös sisältää tietoa tapahtumista, ongelmista, tiedostetuista virheistä, muutoksista sekä julkaisuista.

Tapahtumanhallinta tarvitsee mittareita, joilla voidaan tarkastella sen toimivuutta ja tehokkuutta. Mitattavia asioita voi olla esimerkiksi tapahtumien kokonaismäärä, merkittävien tapahtumien määrä, tapahtuman keskivertokustannus, väärin kirjattujen/kategorisoitujen tapahtumien määrä, Palvelupiste henkilön tapahtumien määrä tai tapahtumamallien mukaan ratkaistut tapahtumat. Tapahtumien hallitsemisessa on haasteita ja riskejä, joita on mainittu ITIL:n kirjassa. Haasteita ovat mm. tapahtumien varhainen havaitseminen, tiedon saatavuus sekä integraatiot asetustenhallintajärjestelmään ja SLM:ään. Riskeinä nähdään tapahtumien pitkät ratkaisuaajat johtuen epäpätevistä tai liian vähäisistä resursseista, tiedon puuttuminen integraatioiden epäonnistumisesta tai epäpätevistä työkaluista sekä toimintojen ja päämäärien erilaiset tavoitteet johtuen huonosti suunnitelluista tai olemattomista OL/UC-sopimuksista. (ITIL 2007, 53-55.)

7.2 Palvelupyynnöprosessi

Palvelupyynnöllä (Service Request) tarkoitetaan yleisesti kaikenlaisia pyyntöjä, jotka ohjautuvat käyttäjiltä IT-osastolle. Useimmat palvelupyynnot ovat pieniä muutoksia kuten salasanan muuttamista, ohjelman asennus tai kysely jostakin asiasta. Palvelupyynnöprosessilla tarkoitetaan näiden palvelupyyntöjen hallitsemista erilaisilla toimenpiteillä. Palvelupyynnöprosessin tavoitteena on tarjota käyttäjille yhteydenottokanava, jonka kautta he voivat pyytää, tilata ja vastaanottaa standardipalveluja ja -laitteita, saada informaatiota sekä apua palautteisiin ja valituksiin. Palvelupyynnöprosessi tuo arvoa liiketoiminnalle siten, että sen kautta liiketoiminta saa nopeasti pääsyn palveluihin ja tätä kautta pystyvät tehostamaan omaa toimintaansa liiketoiminnan hyväksi. Palvelupyynnöprosessi vähentää byrokratian osuutta uusien palveluiden käyttöönottamisessa sekä oikeuksien saamisessa sekä auttaa hallitsemaan palveluita. (ITIL 2007, 55.)

Palvelupyynnöissä kuten tapahtumanhallinnassakin voidaan ottaa käyttöön erilaisia malleja. Palvelupyynnömallit on etukäteen määritelty ja antaa johdonmukaisen tavan käsitellä palvelupyyntöjä. Palvelupyynnömallit ovat myös yleensä muutoksenhallinnan hyväksymiä eivätkä ne muuten eroa tapahtumamallista. Palvelupyynnöprosessiin kuuluu erilaisia aktiviteetteja, metodeja sekä tekniikoita. Palvelupyynnössä voidaan käyttää apuna esimerkiksi Internet-lomakkeita, jossa käyttäjä täyttää lomakkeen ja valitsee tarvittavat kategoriat ja lähettää sen sitten eteenpäin jolloin palvelupyyntö menee automaattisesti prosessin ratkaistavaksi. Jotkin palvelupyynnot vaativat esimiehen hyväksynnän, kuten esimerkiksi pääsyoikeudet tai laitetilausta koskevat palvelupyynnot. Näissä tapauksissa esimiehen hyväksyntä tulisi lisätä palvelupyyntöön ja jos kyse on rahaa vaativasta tilauksesta, esimiehelle tulisi ilmoittaa mahdolliset kustannukset ja palvelupyyntöön tulisi lisätä myös mahdollinen laskutusosuus, jotta sen voi sulkea. (ITIL 2007, 55-57.)

Palvelupyynnön ratkaisu voi tapahtua palvelupisteessä tai palvelupyyntö voidaan ohjata jollekin tietylle ryhmälle tai ulkopuoliselle toimittajalle, koska ratkaisu riippuu aina palvelupyynnön luonteesta. Kun palvelupyyntöön on löytynyt ratkaisu, se palautetaan takaisin palvelupisteeseen joka sulkee sen. Palvelupisteen tulee aina varmistaa, että käyttäjä on tyytyväinen ennen kuin palvelupyyntö suljetaan. Palvelupyyntöprosessi linkittyy useasti eri toimintoihin, kuten tapahtuman-, julkaisun-, omaisuuden- sekä asetushallintaan. Monet palvelupyynnot tulevat palvelupisteen kautta ja ne käsitellään kuitenkin yleensä tapahtumanhallinnassa, mutta jos palvelupyyntö koskee esimerkiksi uuden ohjelman hankkimista, julkaisunhallinnan täytyy ensin määritellä, rakentaa ja testata se ja tämän jälkeen julkaista se sitä haluaville. Uuden ohjelman julkaiseminen vaatii muutostenhallintajärjestelmän tietojen päivityksen ja omaisuudenhallinnassa lisenssien tarkistamisen ja päivittämisen. (ITIL 2007, 57.) Edellä mainittu esimerkki kuvastaa siis hyvin sitä, miten eri toiminnot liittyvät palvelupisteeseen eikä palvelupiste voi vastata aivan kaikesta. Myös joissakin yrityksissä palvelupyöntöprosessi ja tapahtumanhallinta on yhdistetty, eli palvelupiste vastaa palvelupyöntöprosessista sekä tapahtumanhallinnasta.

Palvelupyöntöprosessi tarvitsee tietoja palvelupyöntöjen selvittämiseksi. Tietoja tarvitaan esimerkiksi palvelupyynnöistä, joista pitää selvittää kyseessä olevan palvelun nimi, pyytäjän ja valtuuttajan nimi, mitä prosessia käytetään pyynnön ratkaisuun, kenelle pyyntö oli kirjattu ja mitä tehtiin. Palvelupyöntöprosessi tarvitsee tietoja myös muutoksien pyytämistä (request for change), palveluluettelosta sekä tietoturvapoliitikoista. Palvelupyöntöprosessia voidaan mitata esimerkiksi palvelupyöntöjen määrällä, pyyntöihin kuluneella ajalla, käyttäjätyytyväisyyskyselyllä tai määrällä paljon pyyntöjä on suoritettu määritetyssä ajassa. Tässäkin prosessissa on haasteita ja riskejä, kuten selkeiden ohjeiden määrittely palvelupyöntöjen erottamiseksi tapahtumista ja muutoksista sekä haasteena myös toteuttaa käyttäjille itsepalveluna palvelupyynnön täyttö alusta loppuun. Riskeinä mainitaan huonosti suunniteltu rajaus, jolloin henkilöt eivät tiedä mikä prosessi pyynnön hoitaa, huonosti suunniteltu Internet-lomake käyttäjille, jolloin käyttäjät eivät osaa tehdä pyyntöjä sekä puutteelliset kapasiteetit valvontaan, jolloin tarkkoja mittaustuloksia ei saada kerättyä. (ITIL 2007, 57-58.)

8 Kehitystoimenpiteet

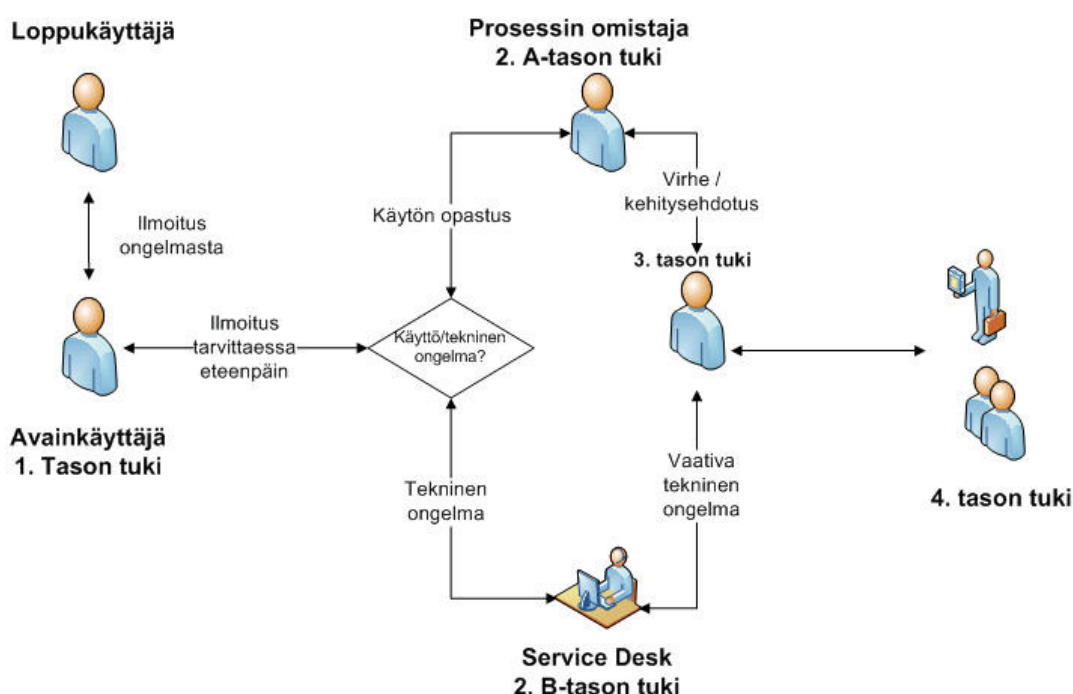
Seuraavaksi käydään läpi työn kehitystoimenpiteet ja tuotokset, joita on käytetty tulosten saavuttamiseksi sekä joita tullaan ottamaan käyttöön tämän työn ohessa sekä tämän työn päättymisen jälkeen.

8.1 Järjestelmän tuki

Järjestelmän palvelupisteen osalta kehitettiin uusi tukiprosessi, tukipyyntöjen käsittelyprosessi, ohjeistukset järjestelmän yleisiin teknisiin ongelmiin sekä kysymyslomake tukipyyntöjen eteenpäin välittämiseksi.

8.1.1 Tukipyyntöprosessi

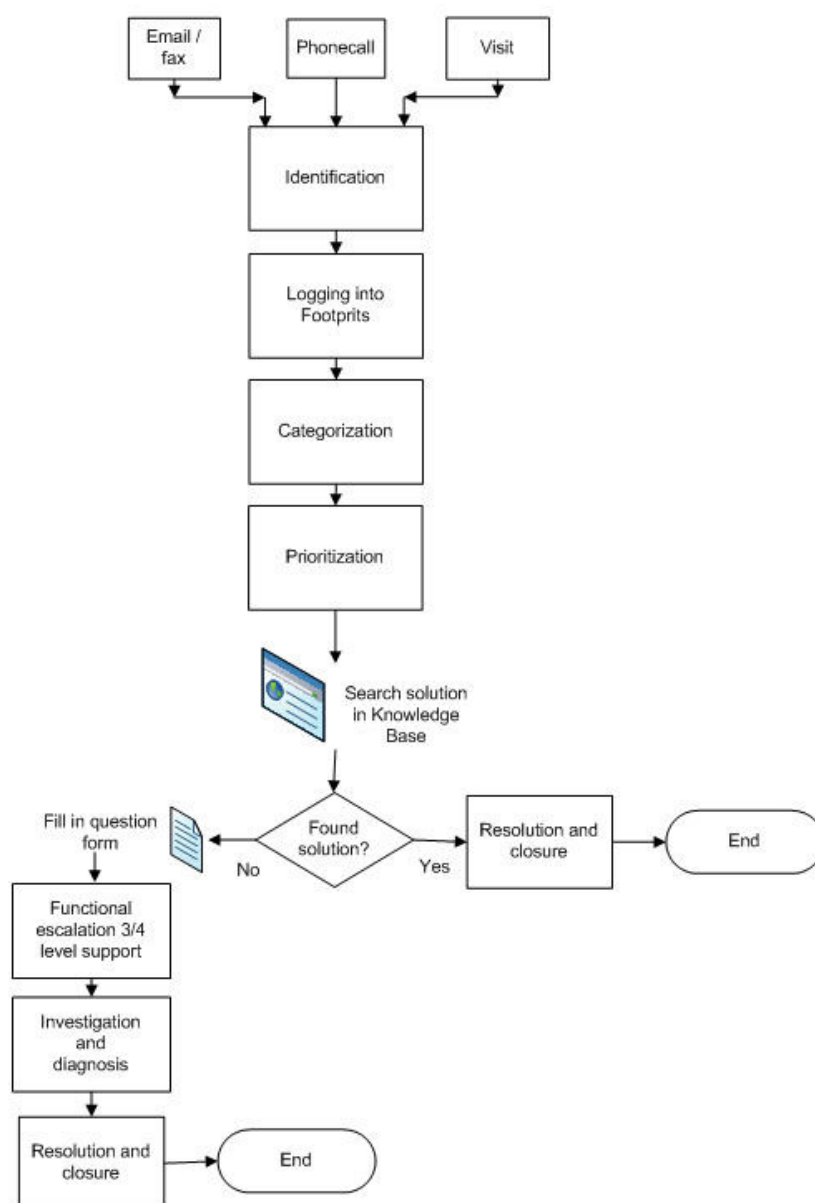
Järjestelmän tukeen kehitettiin kuvion 13 mukainen tukipyyntöprosessi. Tämä tukikuvio julkaistiin alkuvuodesta 2011, joka päivitettiin yrityksen intranettiin järjestelmän sivuille sekä siitä informoitiin yrityksen tietohallinnon uutiskirjeen mukana, joka lähetettiin kaikille Yrityksen suomalaisille työntekijöille.



Kuvio 13. Tukipyyntöprosessi

8.1.2 Tukipyyntöjen käsittelyprosessi

Järjestelmän tukipyyntöjen käsittelyyn suunniteltiin prosessi (kuvio 14), koska järjestelmän tukipyyntöjen käsittelemiseen ei ollut aiempaa prosessikuvaa. Tässä prosessissa on yhdistetty ITIL:n mukaiset palvelupyyntöprosessi sekä tapahtumanhallinta, koska yrityksen palvelupiste käsittelee nämä molemmat heidän käytössään olevan järjestelmän kautta. Palvelupisteen käytössä oleva tukipyyntöjärjestelmä tarjoaa kuitenkin mahdollisuuden erotella palvelupyynnöt tapahtumista, mutta nämä on käsitelty prosessissa yhtenä tapahtumana, koska niiden käsittely ei käytännössä eroa järjestelmän osalta.



Kuvio 14. Tukipyyntöjen käsittelyprosessi ITIL:n teorian pohjalta

Tunnistusvaiheessa tapahtuma tunnistetaan käyttäjältä tulevan ilmoituksen avulla, jonka jälkeen tapahtuma kirjataan tukipyyntöjärjestelmään. Tukipyyntöjärjestelmässä tapahtumalle valitaan asianmukaiset kategoriat ja tässä vaiheessa tapahtuma kirjataan joko palvelupyynnöksi tai tapahtumaksi. Kirjaamisvaiheessa täytetään myös loppukäyttäjän tiedot eikä tapahtumaan pysty tallentamaan ilman asianmukaisia yhteystietoja. Tämän jälkeen tapahtuma priorisoidaan ja etsitään tukipyyntöjärjestelmän ohjetietokannasta ratkaisua. Mikäli ratkaisu löytyy tietokannasta, tapahtumaan kirjataan ratkaisu ja se suljetaan. Jos ratkaisua ei löydy tietokannasta, palvelupiste täyttää kyselylomakkeen ja siirtää tapahtuman ratkaistavaksi seuraavalle tukitasolle. Seuraava tukitaso selvittää ratkaisun ongelmaan ja kun ratkaisu on löytynyt, ongelma korjataan ja informoidaan loppukäyttäjää, jonka jälkeen tapahtuma suljetaan.

8.1.3 Kyselylomake

Kyselylomake (Liite 2) kehitettiin palvelupisteen tueksi järjestelmään liittyvien tukipyyntöjen kirjaamiseen, jotta kaikki oleellinen tieto tulisi kirjattua tukipyyntöjärjestelmään. Kyselylomakkeen on tarkoitus olla yhtenä osana tukipyyntöjen käsittelyprosessia. Kyselylomakkeella pyritään rajaamaan ongelma mahdollisimman pienelle alueelle, jotta vältetään turha ajan käyttöä ongelman tutkimiseen sekä nopean ratkaisun saamiseksi ongelmaan. Kyselylomake on aluksi tarkoitus tallentaa palvelupisteen käyttämään tukipyyntöjärjestelmän tietokantaan ja ohjeistaa palvelupistettä käyttämään kyseistä lomaketta järjestelmän tukipyyntöjä kirjatessa. Kyselylomake on tarkoitus tallentaa myös intranettiin järjestelmän omille sivuille, josta löytyy myös järjestelmän käyttöohjeet sekä tukihenkilöt, jotta asiakkaat voivat ottaa sen kautta yhteyttä palvelupisteeseen.

8.1.4 Yleiset ohjeistukset

Järjestelmän yleisimpiin tukipyyntöihin tehtiin ratkaisumalleja, jotka ovat tallennettu palvelupisteen käyttämään tukipyyntöjärjestelmän tietokantaan. Ohjeistuksista on informoitu palvelupistettä, jotta tukipyyntöjen ratkaisuaika saataisiin pienemmäksi, kun palvelupiste ei kierrätä tukipyyntöjä 3. tason kautta. Mikäli 3. tukitasolle välitetään tukipyyntö, johon löytyy ohjeistus tietokannasta, ratkaistaan tukipyyntö 3. tukitasolla mutta palautetaan tukipyyntö sen jälkeen palvelupisteeseen ja informoidaan ohjeen löytymisestä tietokannasta. Ohjeistukset koskevat lähinnä järjestelmään kirjautumista, järjestelmän eri toimintoihin kuuluvia ominaisuuksia sekä käyttöoikeuksiin liittyviä ohjeita. Ohjeistuksista vastaa 3. tukitason henkilöt, jotka päivittävät ohjeita sekä luovat uusia ohjeita tarpeen mukaan.

8.2 Pienkehitys

Pienkehitykseen vertailtiin kahta eri Agile-menetelmää, joista käydään läpi seuraavaksi niiden hyvät ja huonot puolet. Tämän jälkeen on pohdittu menetelmien ominaisuuksia, joista voisi olla hyötyä järjestelmän pienkehitystehtävissä.

8.2.1 Kanbanin ominaisuuksien arviointi

Kanbanin hyvinä puolina ovat, että se on joustava ja antaa oikeastaan kaiken päätäntävällän tiimin jäsenille. Se ei määrittele tehtävien määrää eri vaiheissa eikä sarakkeiden määrää vaan tiimi saa itse päättää ne. Tämä antaa tiimille itsevarmuutta ja tämän kautta tehtävät voidaan kokea mielekkäämpinä, kun tekijät saavat itse päättää oman työtahdin ja työtavat.

Huonoja puolia Kanbanissa voivat olla osittain samat asiat, jotka ovat myös menetelmän hyviä puolia. Kun menetelmä ei määrää esimerkiksi työn alla olevien tehtävien määrää, tiimin täytyy vain testata mikä heille on sopiva määrä ja siksi menetelmän käyttöönotto voi viedä paljon aikaa. Tiimin henkilöiden täytyy siis tuntea omat osaamisalueensa hyvin, jotta he osaavat määritellä tehtävien keston mahdollisimman tarkasti. Myös huonona puolena menetelmässä on se, että muut tiimin jäsenet saattavat joutua odottamaan muita, jos tehtävämääritykset eri taulun sarakkeissa eivät ole oikeassa suhteessa muihin sarakkeisiin. Eli jos työn alla olevien tehtävien määrä on määritelty liian pieneksi, niin osa tekijöistä on toimeettomana siihen asti kunnes tämä yksi tehtävä on saatu seuraavaan tilaan. Miten tällaisessa tilanteessa saadaan tiimin jäsenet noudattamaan tehtävätaulun määrityksiä, jotteivät tiimin jäsenet määrittelystä huolimatta ala työstämään uutta tehtävää vaikka edellistä ei ole saatu seuraavaan vaiheeseen.

8.2.2 Scrumin ominaisuuksien arviointi

Scrumissa hyvinä puolina järjestelmän ylläpidon näkökulmasta on esimerkiksi se, että se korostaa liiketoiminnan vastuuta järjestelmän kehittämisestä. Laittamalla liiketoiminta tuotteen omistajan rooliin varmistutaan siitä, että oikeita asioita kehitetään liiketoiminnan kannalta oikeaan suuntaan. Sprint soveltuu myös hyvin järjestelmän ylläpitoon, koska se suojelee ylläpitotiimissä olevia henkilöitä eli tässä tapauksessa tiimin jäseniä lisätehtäviltä, koska hyvin useasti tuotteen omistajilta tulee uusia kehittämisasioita kesken sprintin. Eli kun sprintin aikana tuotteen omistajat eli liiketoiminta on tehnyt tiimille tehtävälistan, tuotteen omistajat samalla tiedostavat, että he voivat laittaa esiin tulleet kehitysasiat vasta seuraavan sprintin tehtävälistaan. Hyvinä puolena Scrumissa on myös se, että sprint sisältää uusien ominaisuuksien testauksen. Testaamalla ominaisuudet ennen tuotantoon viemistä vähennetään järjestelmän tukipyynnöjä huomattavasti, kun ominaisuudet tiedetään toimiviksi eikä yllättäviä

järjestelmävirheitä esiinny. Tukipyyntöjen vähentämiseksi tarvitaan kuitenkin suunnitellun testaamisen lisäksi käyttäjien koulutusta tai vähintään tiedottamista uusista ominaisuuksista, jotta ne eivät tule yllätyksenä käyttäjille. Scrum Master on myös yksi hyvä ominaisuus, koska sen avulla saadaan tekijät keskittymään omiin tehtäviinsä, kun Scrum Master poistaa häiriötekijät tiimin työskentelyn tieltä. Scrum Master toimii tavallaan tiimin ja tuotteen omistajien välissä sekä kommunikoi molempien kanssa, mutta tiimi ja tuotteen omistajat eivät saisi kommunikoida keskenään, koska silloin tuotteen omistajat saattavat tottua olemaan yhteydessä suoraan tiimiin kommunikoimatta Scrum Masterin kanssa.

Huonoina puolina Scrumissa on, että siinä tehdään järjestelmään uusia versiota ja Sprintin pituus on menetelmän yleisimpien määritysten mukaan vain 1-4 viikkoa, joten järjestelmän päivitystä tuotantoon tehtäisiin tämän mukaan mielestäni turhan useasti. Järjestelmän tiheä päivitystahti on myös huono ominaisuus, koska käyttäjät joutuisivat liian useasti opettelemaan uusia toimintoja ja osallistumaan koulutuksiin, jotta uusista ominaisuuksista olisi hyötyä. Lisäksi huonona puolena Scrumissa voidaan nähdä tämän kyseisen toiminnanohjausjärjestelmän kannalta myös se, että Scrumin mukainen toiminta vaatii enemmän resursseja mitä tällä hetkellä kyseisen järjestelmän ylläpidossa on.

8.2.3 Yhteenvedo menetelmien soveltuvuudesta pienkehitykseen

Scrum ja Kanban eivät välttämättä aivan sellaisenaan sovellu tämän kyseisen järjestelmän pienkehitykseen, mutta näistä molemmista menetelmistä löytyy hyviä puolia, joita mielestäni voitaisiin ottaa käyttöön pienkehitystiimin päivittäiseen toimintaan.

Pienkehitystiimin toimintaan voitaisiin soveltaa Scrumin sprinttejä eli kehitysjaksoja siten, että yksi kehitysjakso kestäisi aina yhden kvartaalin eli kolme kuukautta. Jokaisen kvartaalin jälkeen julkaistaan järjestelmästä uusi versio, joka sisältäisi prosessin omistajien priorisoimat korjaukset ja uudet toiminnallisuudet. Näistä uusista toiminnallisuuksista viestiminen sekä kouluttaminen kuuluvat prosessin omistajien vastuulle. Scrumista voitaisiin hyödyntää myös tehtävälistaa eli kun tehtävät on priorisoitu tehtäväksi aina tietyn kehitysjakson aikana, tehtävälistalle ei lisätä tämän jälkeen tehtäviä, vaan uudet tehtävät menevät automaattisesti seuraavan kehitysjakson tehtävälistalle priorisoitavaksi. Mikäli kehitysjakson aikana kuitenkin tulee kiireellinen korjaus, tuotteen omistajan on päätettävä minkä tilalla se tehdään, eli tarkoittaen, että kehitysjakson tehtävälistaa ei kasvateta vaan sinne voidaan tarvittaessa vaihtaa tehtäviä. Scrumista voitaisiin ottaa myös päivittäiset palaverit käyttöön. Palaverissa käytäisiin tehtävälistan tilanne ja esiin tulleet ongelmat sekä yhtenä päivänä viikosta tuotteen omistajat osallistuisivat palaveriin. Näin tuotteen omistajat näkisivät tehtävälistan tilanteen sekä saisivat samalla tiedon testattavista asioista.

Kanbanista voitaisiin hyödyntää sen WIP-rajoitusta, eli työn alla olevien tehtävien rajoitusta. Tämän avulla pystytään hallitsemaan eri vaiheissa olevien tehtävien määrää, jottei mihinkään työvaiheeseen kasaudu liikaa tehtäviä. Työn alla olevien tehtävämääritysten avulla saataisiin myös tieto, mikäli jossakin työvaiheessa on ongelmia, ja ne voitaisiin ratkaista nopeammin.

Alla olevassa listassa on tiivistettynä ominaisuuksia Agile-menetelmistä, joita voitaisiin hyödyntää kyseessä olevan toiminnanohjausjärjestelmän ylläpitotehtävissä. Listauksen jälkeen on tehty myös toinen listaus muista ominaisuuksista, joista voisi olla hyötyä ylläpitotoiminnassa.

- kehitysjakso (sprint), pituus 3 kk (tällä hetkellä päivityksiä saatetaan viedä miten satuu)
- tehtävälista, jolle tuotteen omistajat valitsevat maksimissaan neljä tehtävää omalta osa-alueelta (maksimimäärällä järkeistetään järjestelmän muutosten määrää ja järjestelmän liiallista muuttamista sekä taataan jokaisen tuotteen omistajan tehtävien teko ajallaan normaalien resurssien puitteissa)
- WIP-rajoitus tehtävätauluun, jotta saataisiin järkeistettyä tehtävien määrää eri työvaiheissa (vältetään tehtävien kasaantuminen), tehtävät saadaan vietyä koko prosessin läpi oikealla tavalla sekä huomataan mahdolliset ongelmat tarpeeksi ajoissa.
- Scrum Master, joka hoitaisi tiimin ja tuotteen omistajien välisen kommunikoinnin taaten kehittäjille tarpeeksi kattavat tehtävien määritelmät sekä työrauhan ja ajaen samalla liiketoiminnan etua
- minipalaverit 5-15 min, joissa käydään läpi joka aamu työn alla olevat tehtävät ja esiin tulleet ongelmat.

Agile-menetelmistä löydettyjen ominaisuuksien lisäksi voitaisiin ottaa käyttöön seuraavia toimintatapoja

- Tuotteen omistajan sisällyttäminen testausprosessiin, koska tuotteen omistaja loppukädessä tietää parhaiten miten muutoksen pitäisi toimia
- Tuotteen omistajien osallistuminen kerran viikossa tiimin päiväpalaveriin, jotta mahdolliset tehtäviin liittyvät epäselvyydet saadaan yhdellä kertaa selvitettyä sekä tuotteen omistajat pysyvät ajan tasalla tehtävien edistymisestä

8.2.4 Muutospyyntölomake

Pienkehitystä varten suunniteltiin muutospyyntölomake, jonka tarkoituksena on helpottaa kehitysideoiden hallitsemista ja niiden läpikäyntiä, kun ne tulevat samanlaisella lomakkeella ja sisältävät tarvittavat tiedot muutoksen suunnitteluun. Liitteen 3 mukainen muutospyyntö-

lomake on suunniteltu tämän hetkiseen tilanteeseen ja siihen liittyvä jatkokehityssuunnitelma on käsitelty myöhemmin luvussa 11.

Päivämäärä-kenttä on lomakkeessa siksi, että sen avulla pystytään seuraamaan kehitysideoiden elinkaaren pituus sekä tarvittaessa lajittelemaan kehitysideat aikajärjestykseen. Päivämäärä on automaattinen eikä sitä pysty muuttamaan. Muutoksen pyytäjä halutaan tietää, jotta häneen voidaan olla yhteydessä mahdollisten lisätietojen saamiseksi. Pyytäjä voi olla ainoastaan prosessin omistaja, koska heidän vastuullaan on päättää mitä asioita järjestelmää kehitetään. Muutoksen tyyppi voi olla joko korjaava tai uusi ominaisuus. Prioriteetti-valintoina ovat häiritsevä, kiireellinen ja vuorollaan tehtävä. Häiritsevällä tarkoitetaan sitä, että suuri käyttäjäjoukko odottaa muutosta eikä pysty sitä ennen työskentelemään järjestelmän kanssa. Kiireellisellä tarkoitetaan sitä, että sillä ei ole vaikutusta suureen käyttäjämäärään mutta on kuitenkin tärkeä ominaisuus osalle käyttäjiä. Vuorollaan tehtävällä tarkoitetaan sitä, että sillä ei ole oikeastaan muuta kuin käyttäjätyytyväisyyteen liittyvää vaikutusta. Järjestelmän osa-alue täytyy mainita, jotta ylläpitäjät osaavat hahmottaa muutoksen laajuuden. Muutoksen kuvaukseen tulee mahdollisimman tarkka sanallinen kuvaus. Liitetiedosto-kohtaan pyytäjä voi laittaa kuvia tai hahmotelmia muutoksesta taikka kuvia esiintyvistä virheistä. Pyytäjän on myös kerrottava muutoksesta aiheutuvat hyödyt, jotta muutoksia on helpompaa priorisoida keskenään, mikäli muutoksia on paljon samoilla prioriteeteilla ja päivämäärillä.

Vastaanottajalla tarkoitetaan sitä henkilöä, joka ottaa vastaan muutospyyntölomakkeita ja käy niitä läpi yhdessä ylläpitäjien kanssa. Mikäli muutosta ei tehdä, päätös perustellaan ja samalla kerrotaan, mikäli jokin korvaava muutos on tehty tai tehdään. Lomakkeeseen kirjataan muutoksen tekijä, jotta myöhemmin tiedetään kuka muutoksen on tehnyt, jos on esimerkiksi tarvetta samantapaiselle muutokselle tai tarvitaan lisätietoa muutoksesta. Arvioitu työaika ilmoitetaan myös lomakkeessa kustannusten arvioimiseksi. Muutoksen valmistumispäivämäärä laitetaan myös, jotta muutoksen pyytäjä osaa odottaa muutosta oikeaan aikaan. Ratkaisun kuvaukseen tekijä kirjoittaa lyhyen kuvauksen muutoksesta ja dokumentointiosuuteen kirjataan mihin dokumentteihin tehtiin muutoksia.

9 Kehitystyön tulosten arviointi

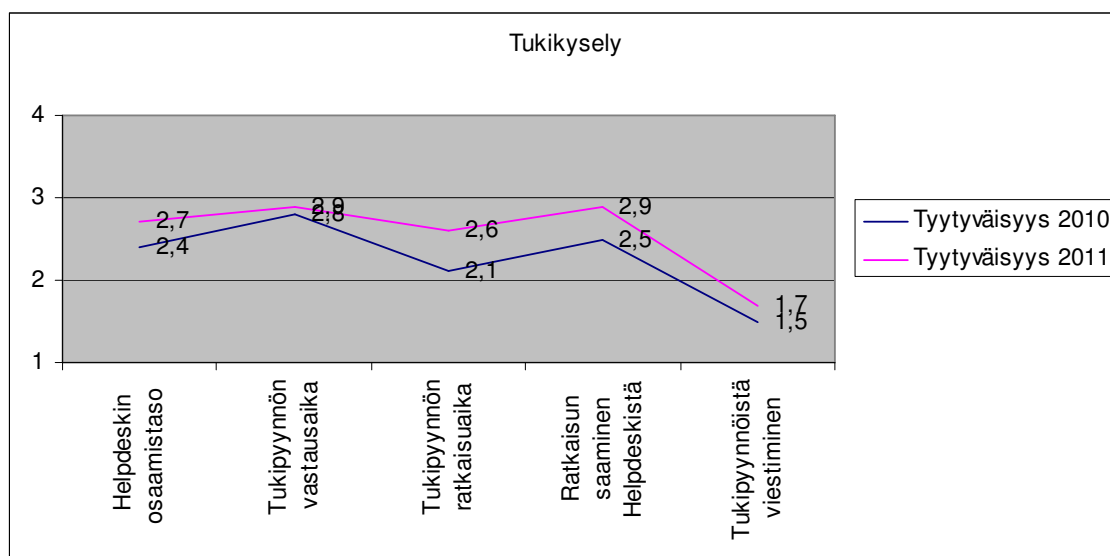
Työn arviointiosuudessa vertaillaan lähtötilannetta nykytilanteeseen eli miten tämä työ on vaikuttanut käyttäjätyytyväisyyteen järjestelmän tukiprosessin osalta. Kehitystyön tuloksia arvioidaan käyttäjätyytyväisyyskyselyn tulosten perusteella. Kyselyn tuloksista on laskettu keskiarvo ja lähtötilanteen tuloksia verrataan nykytilanteen tuloksiin. Tässä osuudessa arvioidaan myös tämän työn onnistumista työn tekijän sekä työ tilaajan näkökulmasta. Työn tekijä arvioi sitä, pääsikö hän työssään siihen lopputulokseen, joka työlle oli asetettu eli parantaa

tuki- ja ylläpitokäytäntöjä sekä käyttäjätyytyväisyyttä. Arviointiosuus sisältää työn tilaajan kommentit jatkokehittämisaieista sekä työn tekijän ammatillisesta kasvusta ja kehittymisestä.

9.1 Käyttäjätyytyväisyyskyselyn tulosten vertailu

Seuraavaksi käydään läpi käyttäjätyytyväisyyskyselyjen tulokset. Aluksi verrataan lähtötilanteen tuloksia nykytilanteeseen (kuvio 15), jolla pyritään antamaan kokonaiskuva kehitystyön tuloksista. Tämän jälkeen käydään vielä kyselyn tulokset läpi kohta kohdalta. Huomiona taulukon lukemiseen (kuvio 15) täytyy muistaa, että viimeisen kysymyksen arvoasteikko poikkeaa muista, eli tukipyyntöistä viestiminen-kysymyksen arvoasteikko on 1-2 kun muiden kysymysten arvoasteikko on 1-4.

Kuviosta 15 voi huomata, että käyttäjätyytyväisyyttä on saatu hieman parannettua mutta kehitettävää on vielä paljon jäljellä. Eniten on saatu parannettua tukipyyntöjen ratkaisuaikaa ja vähiten tukipyyntöjen vastausaikaa. Näihin tuloksiin vaikuttaneita tekijöitä pohditaan jokaisen kysymyksen kohdalla erikseen.

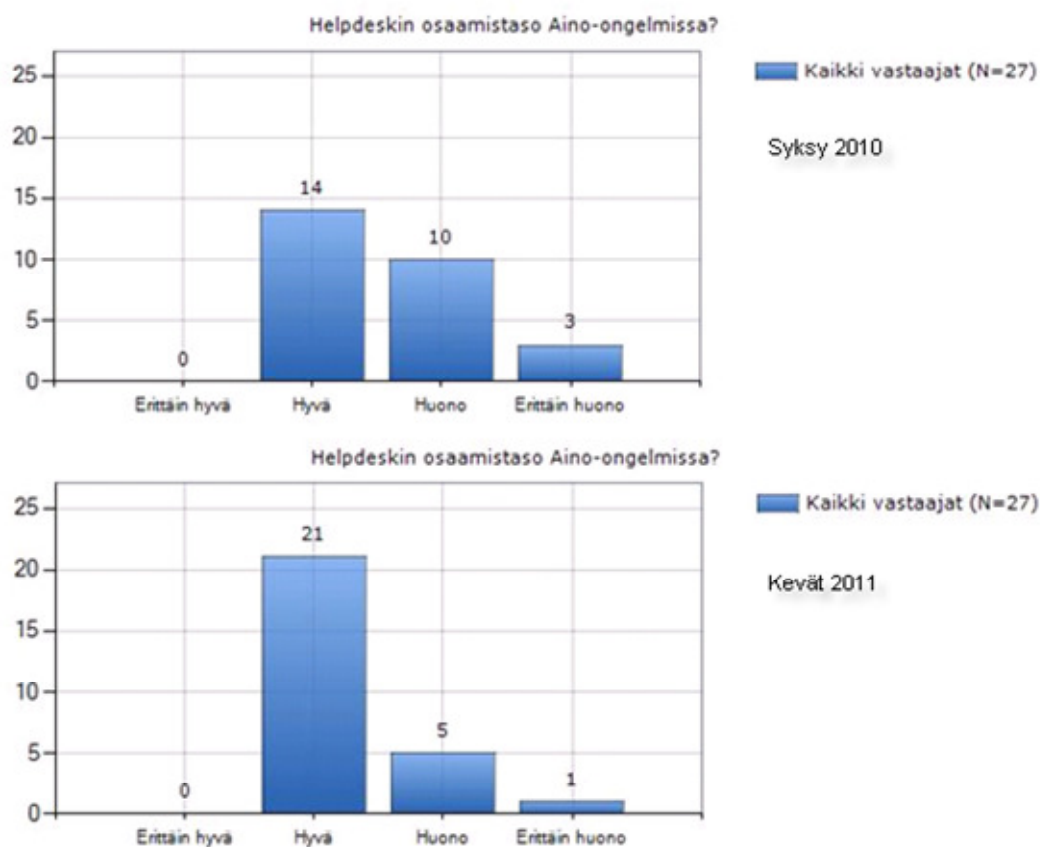


Kuvio 15. Tukikyselyn tulosten vertailu keskiarvon perusteella

Tämän kyselyn pohjalta on hyvä lähteä kehittämään järjestelmän tukiprosessia sekä mitata kehitystyön tuloksellisuutta. Tukiprosessin kehitystä olisi tarkoitus tulevaisuudessa mitata samalla kyselylomakkeella, jotta kyselyn tuloksia pystytään vertailemaan keskenään.

Kuviossa 16 on näytetty tulokset siitä, mitä mieltä kyselyyn osallistujat olivat palvelupisteen (Service Deskin) osaamistasosta. Kuten kuvasta voi huomata, helpdeskin osaamistaso on monen kyselyyn osallistujan mielestä nyt parempi kuin syksyn 2010 tilanteeseen verrattuna. Tämän kyselyn perusteella voidaan olettaa, että tämän opinnäytetyön aikana tehdyillä ohjeistuksilla on ollut vaikutusta helpdeskin osaamistasoon järjestelmän tukipyyntöihin liittyen.

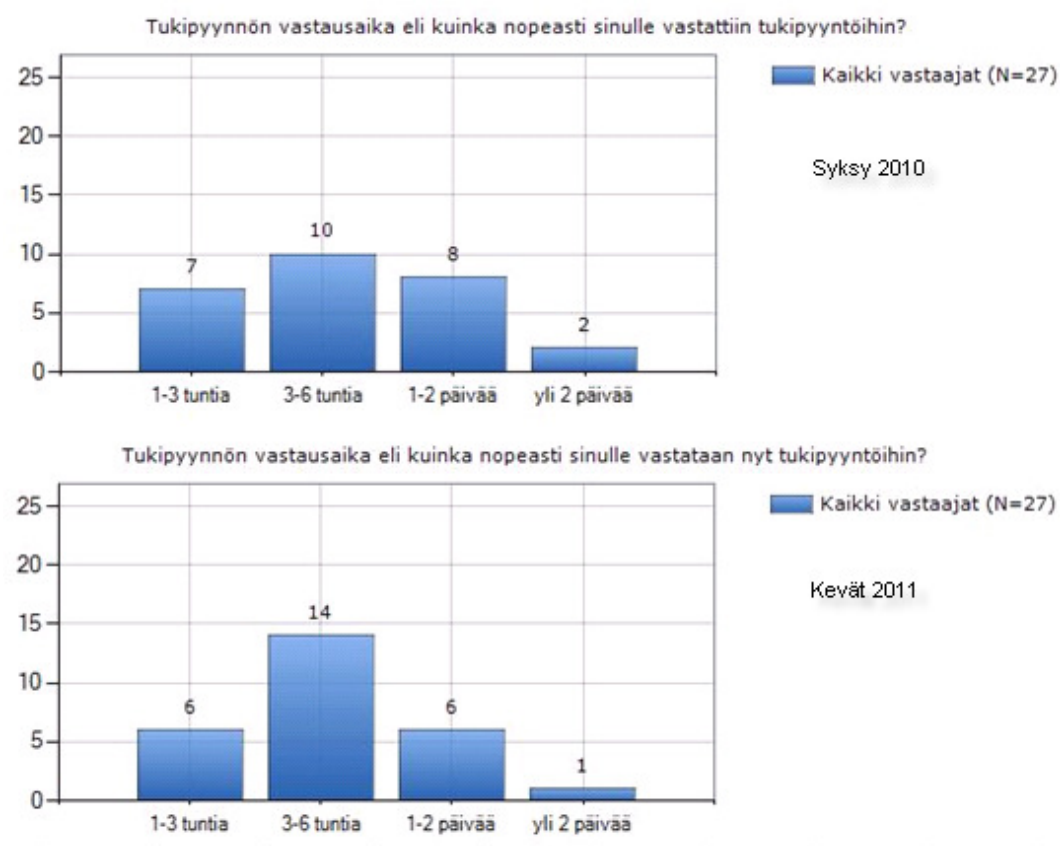
Osaamistasoa voidaan kuitenkin parantaa vielä esimerkiksi viemällä uusia ohjeita tukipyyntöjärjestelmän ohjetietokantaan sekä päivittämällä vanhoja ohjeita ajan tasalle. Myös muistutamalla tietokannassa olevista ohjeista sekä pitämällä pienimuotoisia koulutuksia palvelupisteen henkilöille järjestelmästä, voidaan vaikuttaa osaamistasoon hyvinkin paljon.



Kuvio 16. Kyselyn tulokset, helpdeskin osaamistaso

Kuviossa 17 näytetään tukipyynnön vastausaika - kysymyksen tulokset. Kuten kuviosta huomaa, tukipyyntöjen vastausaikaa on saatu hieman parannettua syksystä 2010, mutta myös joidenkin tapausten kohdalla vastausaika on pidentynyt. Voidaan kuitenkin olettaa, että uuden resurssin lisääminen tukipyyntöprosessiin on vaikuttanut vastausaikaan positiivisesti. Uusi resurssi toimii järjestelmän teknisenä tukihenkilönä järjestelmän tukihenkilöiden sekä palvelupisteen tukena.

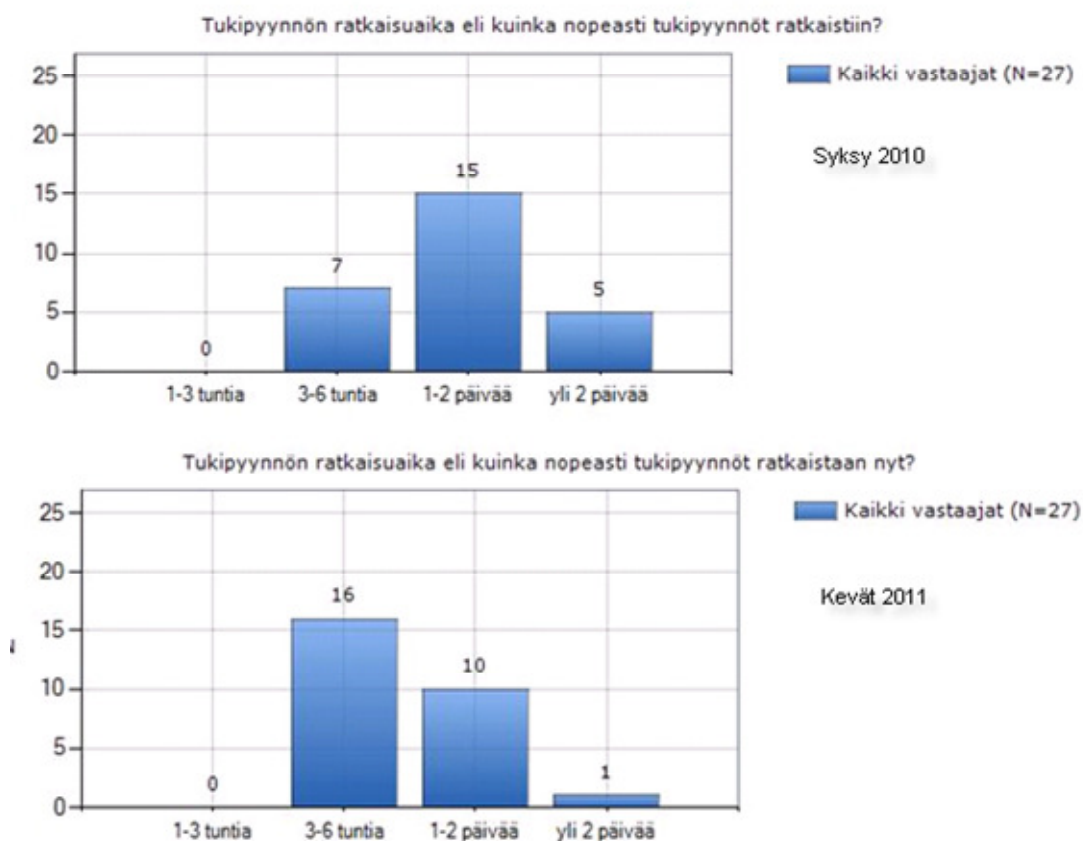
Vastausaika on vieläkin mielestäni liian pitkä ja sitä täytyy parantaa ensimmäisten asioiden joukossa. Vastausaikaa voitaisiin parantaa esimerkiksi tukipyyntöjärjestelmän automaattiviestillä siten, että kun tukipyyntö otetaan vastaan ja kirjataan jollekin henkilölle, asiakkaalle välittyy sähköposti. Sähköposti voisi sisältää esimerkiksi tukipyyntöä käsittelevän henkilön nimen ja sähköpostiosoitteen, tukipyyntönumeron sekä tukipyynnön kuvauksen.



Kuvio 17: Kyselyn tulokset, tukipyynnön vastausaika

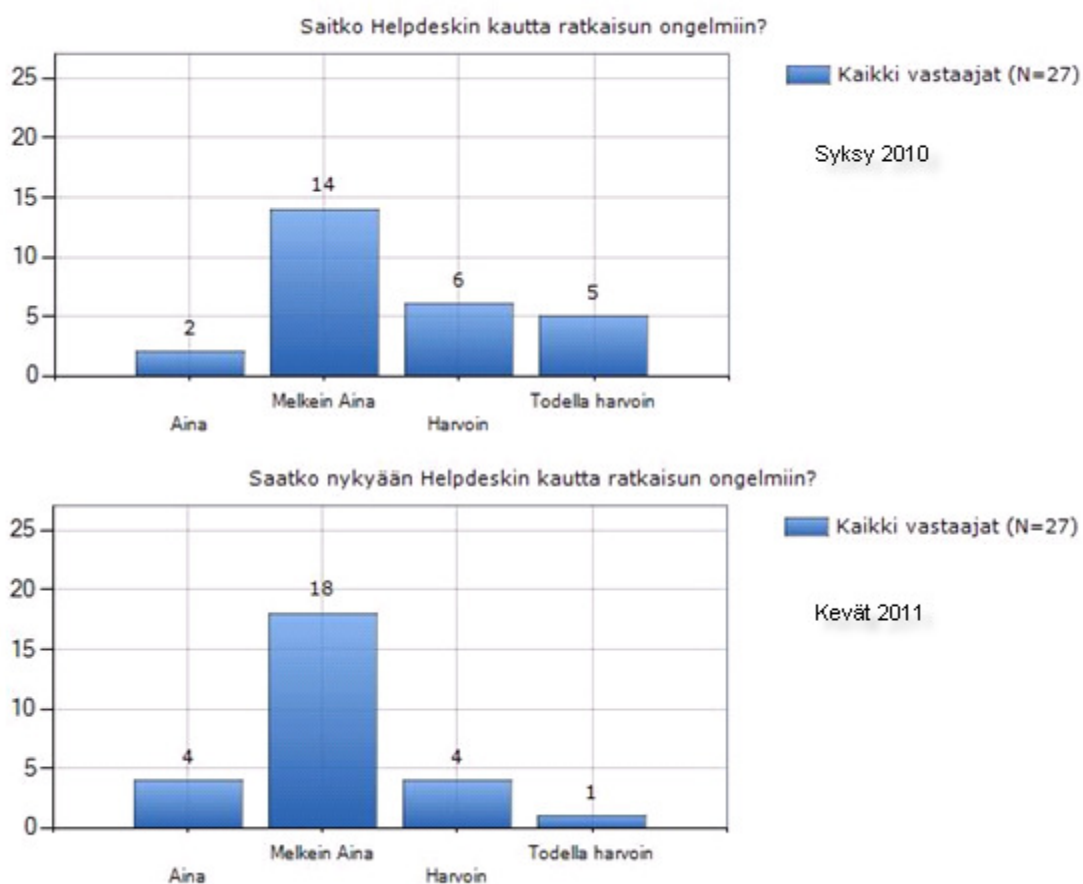
Kuviossa 18 näytetään tulokset tukipyynnön ratkaisuaika -kysymykselle. Kuvasta voi huomata, että ratkaisuaika on lyhentynyt huomattavasti lähtötilanteesta. Mutta vaikka ensimmäisessä kohdassa (1-3 tuntia) ei olekaan yhtään vastausta, tulokset ovat mielestäni silti hyviä, koska yli 2 päivän sekä 1-2 päivän ratkaisuaikoja on saatu pienennettyä huomattavan paljon ja kasvatettua 3-6 tunnin ratkaisuaikaa, joka tulevaisuudessa tulee todennäköisesti olemaan tavoiteratkaisuaika SLA:n määritysten mukaan.

Mikäli tavoite olisi päästä 1-3 tunnin ratkaisuaikaan, tarvittaisiin lisää resursseja ratkomaan järjestelmän päivittäisiä tukipyynnöjä sekä uusia ohjeita täytyisi lisätä tietyin väliajoin ja vanhoja ohjeita päivittää palvelupisteen ohjetietokantaan, jotta ohjeistukset ovat ajan tasalla. Ohjetietokannan avulla palvelupiste pystyisi ratkaisemaan yhä enemmän yleisimpiä tukipyynnöjä ja loppukäyttäjä saisi tällä tavalla nopeammin ratkaisun, koska kolmannen tukitason resurssit eivät keskity ainoastaan tukipyynnöihin vaan myös muihin järjestelmään liittyviin teknisiin tehtäviin.



Kuvio 18. Kyselyn tulokset, tukipyynnön ratkaisuaika

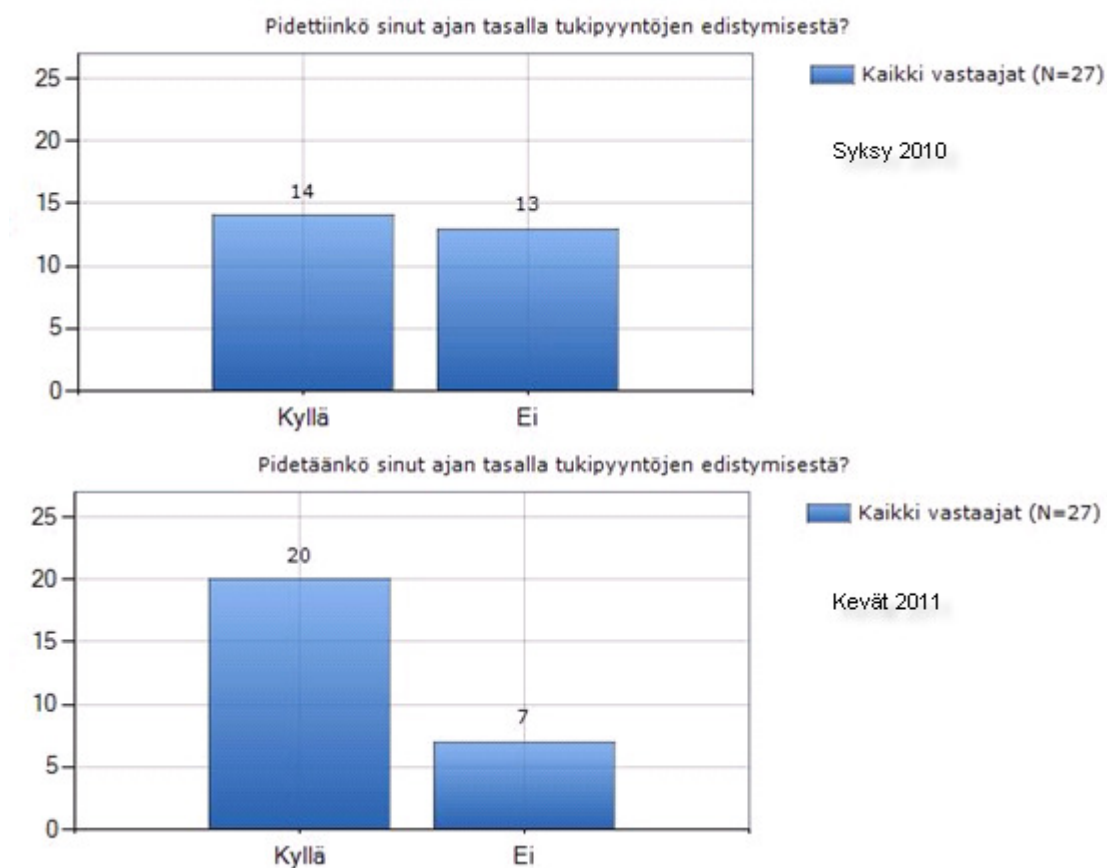
Kuviossa 19 näytetään tulokset kysymykseen, ovatko vastaajat saaneet ratkaisun helpdeskistä. Tästäkin kuvasta voi huomata, että tilanne on nyt parempi kuin syksyllä 2010, kun tukiprosessin kehitys aloitettiin. Kehittämistyön tuloksena voidaan siis sanoa, että tämän työn avulla on saatu kasvatettua tietopääomaa sekä yhteistyötä eri tukitasojen välillä, jonka avulla on saatu tukipyyntöjä ratkaistua helpdeskissä tai helpdeskin kautta useammin kuin syksyllä. Myös voidaan olettaa, että järjestelmän tukihenkilöt ovat alkaneet käyttämään enemmän yrityksen palvelupistettä järjestelmän tukipyyntöjen ratkaisemisessa kun aiemmin tukihenkilöt ottivat suoraan yhteyttä järjestelmän ylläpitäjiin.



Kuvio 19. Kyselyn tulokset, ratkaisun saaminen helpdeskistä

Kuviossa 20 näytetään tulokset kysymykselle pidettiinkö loppukäyttäjä ajan tasalla tukipyynnön edistymisestä. Kuvasta voi huomata, että tätä osa-aluetta tukiprosessissa on myös parannettu huomattavasti syksystä, mutta parannettavaa on vielä paljon, koska tavoitteena on pitää kaikki asiakkaat ajan tasalla tukipyyntöjen edistymisessä.

Loppukäyttäjän ajan tasalla pitämistä voitaisiin parantaa tukipyyntöjärjestelmän automaattiviestillä, samanlaisella viestillä kuin oli kyse tukipyyntöjen vastausaika-kysymyksessä. Tämä voisi toimia esimerkiksi siten, että aina kun tukipyynnön tila vaihtuu tai se siirretään toiselle henkilölle käsiteltäväksi, asiakkaalle välittyy viesti. Viestissä voisi olla esimerkiksi seuraavat tiedot: Tukipyynnön käsittelijän nimi ja sähköpostiosoite, tukipyynnön tila sekä tehdyt toimenpiteet tukipyynnön ratkaisemiseksi.



Kuvio 20. Kyselyn tulokset, tukipyynnöistä informointi

9.2 Työn onnistumisen arviointi

Koen onnistuneeni hyvin tässä työssä. Kuten käyttäjätyytyväisyyskyselyn tulokset osoittavat, olen saanut tämän työn kehitystoimenpiteiden avulla parannettua järjestelmän käyttäjätyytyväisyyttä sekä olen pystynyt vaikuttamaan tukipyyntöjen tehokkaampaan käsittelyyn palvelu-

pisteessä. Kehitystyön aikana järjestelmän tukihenkilöt ovat myös alkaneet käyttämään palvelupistettä tukipyyntöjen eteenpäin viemisessä, mikä on mielestäni yksi tärkeimmistä saavutuksista, koska tällä tavalla pystytään mittaamaan järjestelmään liittyviä palvelupyyntöjä sekä aiheutuneita tapahtumia. Olen myös suunnitellut tulevia muutoksia järjestelmän tuki- ja ylläpitokäytäntöjen parantamiseksi, jotta käyttäjätyytyväisyyttä saadaan parannettua entistään sekä tehostettua tukipyyntöprosessia kaikilla tukitasoilla.

Työn tilaaja on erityisen tyytyväinen opinnäytetyön tekijän järjestelmälliseen ja käytännölliseen työskentelytapaan sekä lopputulosten hyödynnettävyyteen. Lisäksi opinnäytetyön jatkokehittämisajatuksia otetaan yhdessä työn alle ja viedään eteenpäin. Tyytyväisyyskyselystä ja siihen liittyvistä kehittämistoimenpiteistä tehdään osa ylläpidon vuosittaista työskentelyä.

10 Pohdinta

Tässä kappaleessa pohditaan, miten työlle asetetut tavoitteet on saavutettu, mitä tutkimus on antanut, miten tutkimus on auttanut alkuperäisen ongelman selvittämisessä sekä miten työ on edistänyt työn tekijän ammatillista osaamista. Tässä kappaleessa käydään läpi myös työn aikana saadut tuotokset sekä työn ohessa heränneet jatkokehitysajatuksia.

10.1 Johtopäätökset

Työlle asetetut tärkeimmät tavoitteet saavutettiin eli kehitystyön avulla saatiin parannettua käyttäjätyytyväisyyttä, tehostettua tukiprosessia, tuotettua uusi prosessikuva tukipyyntöjen käsittelyyn sekä luotiin uusia lomakkeita järjestelmän tuki- ja pienkehitystehtäviin.

Tämä tutkimus antoi hyödyllistä tietoa järjestelmän käyttäjätyytyväisyydestä järjestelmän tukipyyntöihin liittyen. Tutkimuksen avulla saatiin selville heikoimmat kohdat palvelupisteessä, joita lähdettiin kehittämään. Tutkimuksen avulla saatiin selville myös Agile-menetelmien, Scrumin ja Kanbanin hyvät sekä huonot puolet, joita voidaan mahdollisesti tulevaisuudessa hyödyntää järjestelmän pienkehitystehtäviin.

Kehitystyön avulla selvitettiin käyttäjätyytyväisyyden lähtötilanne tukikyselyllä, jota pyrittiin parantamaan erilaisten toimintojen kautta. Toiminnoilla tarkoitetaan tämän työn aikana luotuja ohjeistuksia sekä uusittua tukiprosessia loppukäyttäjän näkökulmasta, joilla oli erittäin suuri vaikutus tavoitteisiin nähden eli käyttäjätyytyväisyyskyselyn tuloksiin.

Työn kautta onnistuttiin parantamaan kyseisen järjestelmän käyttäjätyytyväisyyttä sekä luomaan tarvittavia ohjeita ja lomakkeita järjestelmän tuessa toimivien henkilöiden tueksi. ITIL-

teoriaan pohjautuen luotiin myös tukipyyntöjen käsittelyprosessi, jollaista yrityksellä ei vielä ollut tämän kyseisen järjestelmän tukipyyntöihin liittyen.

10.2 Oman ammatillisen kehityksen arviointi

Olen onnistunut kehittämään omaa osaamistani järjestelmän tukitehtävissä sekä kartoittanut teorian tietämystäni IT-alalla käytettävistä käsitteistä ja menetelmistä opinnäytetyöni kautta. Olen myös saanut tutustua uusiin menetelmiin kehitystyön ohella, joista koen olevan omassa työssäni sekä oman osaamisen kehittymisen kannalta todella paljon hyötyä. Työn tekeminen on opettanut itsekuria sekä ison projektin hallintaa alusta loppuun asti. Itsenäinen työskentely on myös vahvistunut työn aikana. Työssäni onnistuin mielestäni noudattamaan toiminnallisen tutkimuksen asettamia vaiheita ja rajoituksia, sekä pyrin myös tuomaan nämä esille työssäni.

10.3 Jatkokehittämisaalueet

Työn edetessä ilmeni erilaisia jatkokehitysajatuksia työn tekijän kuten myös käyttäjätyytyväisyyskyselyn perusteella (Liite 1 B), joita käydään lyhyesti läpi.

Järjestelmän pienkehitykseen olisi tarvetta suunnitella prosessi, jonka mukaan ylläpitotöitä hoidettaisiin hallitusti. Prosessissa voitaisiin käyttää hyödyksi Agile-menetelmien ominaisuuksia, joita on kuvattu luvussa 8.2.3. Ylläpitoprosessin käyttöönottamiseksi tulisi myös suunnitella prosessiin osallistuvien henkilöiden sitouttaminen ja motivointi ylläpitoprosessin noudattamiseksi, jotta prosessissa olisi myös hyötyä ja se tuotaisiin käytännön tasolle.

Järjestelmän käyttäjätyytyväisyyskyselyn toteutukseen sekä tulosten hyödyntämiseen tulisi kehittää suunnitelma, jossa mietitään esimerkiksi, miten kyselystä saatuja vastauksia analysoidaan sekä hyödynnetään järjestelmän pienkehityksessä sekä tuessa. Kyselylle tulisi myös suunnitella toteutusaikataulu sekä osallistujalistat siten, että loppukäyttäjätkin pääsevät kertomaan oman mielipiteensä järjestelmän toimivuudesta sekä kehitysehdotuksista. Kyselyn toteuttamisessa tulisi ottaa huomioon järjestelmän pienkehitysprosessin julkaisuaikataulu, jottei kyselyä toteuteta samaan aikaan kun järjestelmään tehdään päivityksiä. Myös kyselystä saatujen tulosten viestintään olisi hyvä suunnitella prosessi, kuinka tulokset kerrotaan kyselyyn osallistujille sekä millä aikataululla.

Tukipyyntöihin suunniteltua kyselylomaketta tulisi myös kehittää, jotta siitä saataisiin enemmän hyötyä järjestelmätuessa ja jotta vähennettäisiin tukipyyntöjen perustietojen saamiseen kulunutta aikaa. Kysymyslomakkeen suunnitteluprosessiin voitaisiin sisällyttää esimerkiksi kyseisen järjestelmän tukihenkilö sekä yrityksen palvelupisteen henkilöitä. Suunnitteluproses-

sissa voitaisiin analysoida aiempia tukipyyntöjä ja niistä saatavien tietojen avulla lisätä lisäksymyksiä kyselylomakkeeseen.

Muutospyyntölomaketta tulisi myös kehittää sen mukaan, miten ylläpitoprosessi kehittyy. Lomakkeen suunnitteluun tulisi sisällyttää järjestelmän tukihenkilö, järjestelmän kehittäjä sekä järjestelmän omistaja, jotta lomake sisältäisi niin käyttäjän kuin teknisen puolen tiedot tarpeeksi kattavasti. Lomakkeen vastuuhenkilö olisi myös hyvä valita, jotta lomaketta koskevat kysymykset ja päivityspyynnot käsitellään järjestelmällisesti.

Lähteet

Airaksinen, T. & Vilkkä, H. 2003. Toiminnallinen opinnäytetyö. Helsinki: Tammi.

Certification.info. 2009. Viitattu 1.4.2011

<http://itsm.certification.info/>

Crisp Ab. Lean/Kanban. Viitattu 13.3.2011

<http://www.crisp.se/kanban>

Fehér, P. 2009. ITIL v3 Structure. Krpm. Viitattu 26.3.2011

<http://krpm.files.wordpress.com/2009/01/peter-feher-til-v3-structure-krmpwordpresscom.pdf>

Harsu, M. 2003. Ohjelmien ylläpito ja uudistaminen. Helsinki: Talentum.

Harsu, M. 2010. Ohjelmien ylläpito ja evoluutio. TTY Ohjelmistotekniikka. Viitattu 26.2.2011

<http://www.cs.tut.fi/~evo/kalvot/yllapito6.pdf>

Harsu, M. 2010. Ohjelmien ylläpito ja evoluutio. TTKK Ohjelmistotekniikka. Viitattu 7.3.2011

<http://www.cs.tut.fi/~evo/kalvot/tentti6.pdf>

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2000. Tutki ja kirjoita. 6.-7. painos. Vantaa: Tummavuoren kirjapaino Oy.

Holopainen, M. & Pulkkinen, P. 2003. Tilastolliset menetelmät. 1.-2. painos. Vantaa: Dark Oy.

ITIL. 2007. Service Operation. Englanti: The Stationery Office.

ItSMF. ITIL kirjasto. Viitattu 26.3.2011

<http://www.itsmf.fi/itilkirjasto>

Kananen, J. 2009. Toimintatutkimus yritysten kehittämisessä. Jyväskylä: Jyväskylän ammatti-korkeakoulu.

Kautto, M. 2009. Palvelutasosopimukset. Helsingin yliopisto. Viitattu 20.2.2011

http://www.cs.helsinki.fi/group/cinco/teaching/2009/soc-seminaari/abstracts/kautto_abstract.pdf

Kniberg, H. 2009. Kanban vs Scrum. Crisp AB. Viitattu 1.3.2011

<http://www.crisp.se/henrik.kniberg/Kanban-vs-Scrum.pdf>

Koskela, L. 2007. Scrum - Ketterien menetelmien markkinajohtaja. Reaktor Innovations Oy. Viitattu 1.3.2011

http://ttlry-fi-bin.directo.fi/@Bin/cd3c2cdf7e10e9a65118fe34636303a2/1298988807/application/pdf/11062393/04_ScrumMarketLeaderOfAgileMethods_handout_LasseKoskela.pdf

Koto, A. 2009. Tietojärjestelmien väliset rajapinnat sähköjakeluverkon käyttötoiminnassa. Tampereen teknillinen yliopisto. Viitattu 23.2.2011

http://docs.google.com/viewer?a=v&q=cache:gRdETchXY-sJ:webhotel2.tut.fi/units/set/raportteja/dg/adine/Diplomityo_Koto_Antti.pdf+tietoj%C3%A4rjestelm%C3%A4ll%C3%A4+tarkoitetaan&hl=fi&gl=fi&pid=bl&srcid=ADGEESiKMy5vKNGkF6F-pkObuFWUPBYc-mo3J3SCTmSXt33EozzBEjfV4XWEUzhAyd12ibvLAqvQJVhR0ZuW7ldF2rj5wY557N05K84AXo3wJZF9Sf1PbGU4SNWfaEmzMCUrOlyKPkgQ8&sig=AHIEtbRpZ95JiZrP-b5BY1anZp6ALFsicw

Kupiainen, M. 2007. Pk-yrityksen toiminnan kehittäminen ITIL-mallin avulla. Joensuun yliopisto. Viitattu 26.3.2011

ftp://cs.joensuu.fi/pub/Theses/2007_MSc_Kupiainen_Mikko.pdf

Laamanen, K. & Tinnilä M.. 2009, 4. uudistettu painos. Prosessijohtamisen käsitteet. Teknologia-teollisuus Oy.

Lindström, J. 2006. Scrum. Reaktor Innovations Oy. Viitattu 27.2.2011

<http://www.reaktor.fi/web/fi/teknologia-ja-tutkimus/scrum>

Lukumaa, T. 2006. Agile-menetelmät. Reaktor. Viitattu 23.2.2011

<http://www.reaktor.fi/web/fi/teknologia-ja-tutkimus/agile-menetelmat>

Mäkelä, N. 2004. Ylläpidon tiedontarpeiden lähteet. Helsingin yliopisto. Viitattu 26.2.2011

www.helsinki.fi/~nmmakela/seminaari.doc

Nylander, T & Kirkkala, T. 2008. ITILin hyödyt ja haitat. Tietojärjestelmäprojekti ja muutos. Haaga-Helia ammattikorkeakoulu. Viitattu 15.3.2011

http://myy.haaga-helia.fi/~kalei/tjmuutos/raportit/ITIL_yhteenvedo_final.pdf

Pirinen, E. 2010. Kanban-ohjelmistokehitysmenetelmän vaikutuksista arvonn muodostukseen ja ohjelmistokehittäjän työhön. Helsingin yliopisto. Viitattu 13.3.2011

http://www.softwarefactory.cc/pdf/ProGradu_Pirinen.pdf

Reini, P. 2008. Scrumia vai ei - Case yritys X. Jyväskylän ammattikorkeakoulu. Viitattu 8.3.2011

https://publications.theseus.fi/bitstream/handle/10024/19205/jamk_1238495903_3.pdf?sequence=2

Vilkka, H. 2007. Tutki ja mittaa. Jyväskylä: Tammi.

Kuviot

Kuvio 1. Lähtötilanne tukiprosessin kehittämiseksi syksyllä 2010	8
Kuvio 2. kehitystyön vaiheet	9
Kuvio 3. Tukikysely, lähtötilanne 2010 -tulokset	12
Kuvio 4. Palvelupisten rakenne yrityksessä	13
Kuvio 5. ITIL v3 rakenne (Fehér 2009.).....	16
Kuvio 6. Ylläpidon osa-alueet jaoteltuna	19
Kuvio 7. Esimerkki muutospyyntödokumentista (Harsu 2003, 84).	23
Kuvio 8. Eri menetelmien soveltuvuuden vertailu (Kniberg 2009, 9).	24
Kuvio 9. Kanbanin työpöytä (Kniberg, 2009, 1).	25
Kuvio 10. Scrum-prosessi (Lindström 2006, 2).	26
Kuvio 11. Scrum työvaiheet (Kniberg 2009, 12).	28
Kuvio 12. Tapahtumanhallintaprosessin kulku ITIL:n mukaan.....	30
Kuvio 13. Tukipyyntöprosessi	33
Kuvio 14. Tukipyyntöjen käsittelyprosessi ITIL:n teorian pohjalta	34
Kuvio 15. Tukikysely, tulosten vertailu	40
Kuvio 16. Kyselyn tulokset, osaamistaso	41
Kuvio 17: Kyselyn tulokset, vastausaika	42
Kuvio 18. Kyselyn tulokset, ratkaisuaika.....	43
Kuvio 19. Kyselyn tulokset, ratkaisun saaminen.....	44
Kuvio 20. Kyselyn tulokset, informointi	45

Liite 1 A: Käyttäjätyytyväisyyskysely

Hei ja mukavaa kevään alkua!

Olen tekemässä opinnäytetyötäni järjestelmä x tukeen ja erityisesti Helpdesk osuuden kehittämiseen liittyen. Kysyisin nyt teidän mielipidettänne tältä osin järjestelmän tuen toimivuudesta sekä syksyllä 2010 että nykytilanteessa. Lisäksi olen kiinnostunut myös jatkokehittämisajatuksista (kyselyn loppuosa).

Toivoisin, että teillä olisi hetki aikaa vastata kyselyyni.

Järjestelmän tuen toimivuus **syksyllä marraskuussa 2010**, mitä mieltä olet..

1. Helpdeskin osaamistasosta järjestelmä-ongelmissa?
(1=Huono 2= Kohtalainen 3=Hyvä 4=Erittäin hyvä)
2. Tukipyynnön vastausajasta (kuinka nopeasti sinulle vastattiin tukipyyntöön)?
(1=yli 2pv, 2=1-2pv, 3=2-4h, 4=1-2h)
3. Tukipyynnön ratkaisuaikasta (kuinka nopeasti ongelma ratkaistiin)?
(1=yli 2pv, 2=1-2pv, 3=2-4h, 4=1-2h)
4. Saitko Helpdeskin kautta avun ongelmaan (ratkaisiko helpdesk tai muu taho tukipyynnön)?
(1=en saanut 2=Harvoin 3=Yleensä 4=Aina)
5. Pidettiinkö sinut ajan tasalla tukipyynnön edistymisestä?
(1= Ei 2= Kyllä)

Järjestelmän tuen toimivuus nyt **maaliskuussa 2011**, mitä mieltä olet nyt tilanteesta

6. Helpdeskin osaamistasosta järjestelmä -ongelmissa?
(1=Huono 2= Kohtalainen 3=Hyvä 4=Erittäin hyvä)
7. Tukipyynnön vastausaika (kuinka nopeasti sinulle vastattiin tukipyyntöön)?
(1=yli 2pv, 2=1-2pv, 3=2-4h, 4=1-2h)
8. Tukipyynnön ratkaisuaika (kuinka nopeasti ongelma ratkaistiin)?
(1=yli 2pv, 2=1-2pv, 3=2-4h, 4=1-2h)
9. Saitko Helpdeskin kautta avun ongelmaan (ratkaisiko helpdesk tai muu taho tukipyynnön)?
(1=en saanut 2=Harvoin 3=Yleensä 4=Aina)
10. Pidettiinkö sinut ajan tasalla tukipyynnön edistymisestä?
(1= Ei 2= Kyllä)

Liite 1 B: Käyttäjätyytyväisyyskysely, avoimet kysymykset

Järjestelmä ylläpidon tilanne **tulevan puolen vuoden aikana 2011**, mitä mieltä olet

Keskeisimmät kehittämiskohteet tukipalvelun osalta:

Lopuksi voit antaa koko järjestelmän tukeen liittyen kehitysehdotuksia sekä myös muuta palautetta!

Kehitysehdotukset:

Muu palaute:

Kiitos avustasi ☺ !

Liite 2: Kysymyslomake tukipyyntöjen käsittelyyn

Kysymyslomake

1. Pääseekö käyttäjä kirjautumaan
 - a. Kyllä__
 - b. Ei__
 - i. Virheilmoitus _____

2. Yritys
 - a. Yritys x__
 - b. Ulkoinen__
 - i. Yritys _____
3. Mikä ohjelman osa-alue _____
4. Milloin viimeksi toiminut __. __. __ (pv.kk.vuosi)
5. Onko muilla käyttäjillä sama ongelma
 - a. Kyllä, kaikilla projektin jäsenillä__
 - b. Kyllä, joillakin projektin jäsenillä__
 - i. Käyttäjät _____

 - c. Ei, vain yhdellä käyttäjällä__
6. Ongelman kuvaus/virheilmoitukset (Esim. projektin nimi, jossa ongelma esiintyy, missä vaiheessa ongelma ilmenee)

Liite 3: Muutospyyntölomake

Muutospyyntödokumentti

- | | |
|--------------------------|---|
| 1. Päivämäärä | ____.____.____ |
| 2. Muutoksen pyytäjä | _____ |
| 3. Muutoksen tyyppi | Korjaava__ Uusi ominaisuus__ |
| 4. Prioriteetti | Häiritsevää__ Kiireellinen__ Vähäinen__ |
| 5. Järjestelmän osa-alue | _____ |
| 6. Muutoksen kuvaus: | _____ |
| | _____ |
| | _____ |
| | _____ |
| | _____ |
| 7. Muutoksen hyödyt: | _____ |
| | _____ |
| | _____ |
| | _____ |

Liitetiedosto(t):